

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN:
SISTEMAS DE TELECOMUNICACIÓN



PROYECTO FINAL DE CARRERA

VITAL GYM
MONITORIZACIÓN ECG EN TIEMPO REAL

AUTOR:

PEDRO PABLO RAMOS ROMERO

COTUTOR:

JOSÉ IGNACIO MORENO NOVELLA

AÑO 2012

ÍNDICE

INTRODUCCIÓN.....	iii
VISIÓN GENERAL DEL PROYECTO	iv
ANÁLISIS DE MERCADO.....	v
REQUISITOS DEL SISTEMA.....	vii
CASOS DE USO.....	viii
CASOS DE PRUEBA	ix
DISEÑO Y ARQUITECTURA.....	x
DIAGRAMAS DE ACTIVIDAD	xi
BASE DE DATOS	xii
CÓDIGO Y DOCUMENTACIÓN DE DESARROLLO.....	xiii
<i>ANEXO</i>	<i>xiv</i>

INTRODUCCIÓN

Este proyecto fue realizado entre los meses de Febrero y Junio de 2011, durante mi estancia en la *Universidade de Aveiro* (Portugal) como estudiante del programa de intercambio universitario europeo Erasmus.

Constituye la parte fundamental de la asignatura *Projecto em Engenharia Informática* (de 12 ECTS), correspondiente a la titulación de Ingeniería de Computadores y Telemática, y fue desarrollado por un equipo de 6 estudiantes (entre los cuales me incluyo) bajo la supervisión del Prof. José Maria Amaral Fernandes, coordinador de dicha asignatura.

La función de cotutor del proyecto dentro de la Universidad Carlos III de Madrid ha sido desempeñada por el Dr. José Ignacio Moreno Novella (Dpto. de Ingeniería Telemática), que también fue mi coordinador académico durante mi estancia de nueve meses en Portugal.

El producto diseñado y desarrollado (VitalGym) durante el proyecto está concebido para ser un sistema de monitorización ECG (Electrocardiograma) para atletas que acuden normalmente a gimnasios. Como la forma física de una persona puede ser inferida mediante la respuesta cardíaca al ejercicio físico, el propósito de este sistema es ayudar al gimnasio a establecer un programa de entrenamiento adecuado en función de las necesidades individuales de cada atleta. Por lo tanto, se pretende que los gimnasios abandonen la práctica de crear programas de entrenamientos generalizados para todo el mundo (como ocurre en la actualidad) y opten por establecer programas totalmente personalizados a través del producto en cuestión. Esto contribuirá a la satisfacción del atleta respecto al gimnasio, lo cual incrementará la reputación del gimnasio.

Es posible ver un vídeo de presentación del producto en el siguiente enlace: <http://www.youtube.com/watch?v=NdQXhn11bWo>

A continuación se incluye un resumen del proyecto. Para obtener información más detallada acerca de los diferentes apartados expuestos, se recomienda consultar la memoria completa que se puede encontrar en el *Anexo* (pág. xiv).

VISIÓN GENERAL DEL PROYECTO

Llevamos a cabo una visión generalizada del proyecto, teniendo en consideración los siguientes aspectos:

- **Planteamiento del problema:** La mala interpretación de los resultados obtenidos durante el desarrollo de las sesiones de entrenamiento en los gimnasios hace difícil conocer las limitaciones físicas de los atletas. Por lo que la ausencia de un programa de entrenamiento personalizado afecta tanto a al atleta como a la credibilidad del gimnasio, ya que sólo se satisface a la “población promedio”. Una solución a este problema sería una aplicación para monitorizar el pulso cardíaco y procesar los datos recibidos, lo cual permitiría evaluar la forma física del atleta y proporcionar las bases para la creación un entrenamiento personalizado de forma precisa.
- **Posicionamiento del producto:** VitalGym estará destinado a los gimnasios que quieran mejorar sus programas de entrenamiento sin tener que contratar a un gran número de entrenadores personales, proporcionando a éstos, información acerca de la condición física del atleta. Se comercializará en el mercado como una herramienta para los gimnasios. Ya que los atletas podrán seguir su progreso personal, será el primer producto orientado al gimnasio y al atleta al mismo tiempo.
- **Descripción de las partes interesadas:** El *Personal Trainer* podrá ajustar el programa de entrenamiento de cada atleta a través de la información proporcionada por el sistema. El *Atleta* será monitorizado y podrá comprobar su propio progreso y las indicaciones del entrenador. El *Administrador* gestionará todo el sistema y será el responsable de añadir nuevos entrenadores y asignarlos a sus atletas correspondientes.
- **Descripción del producto:** La necesidad prioritaria es la monitorización ECG, que será cubierta mediante una fuente de datos ECG y la comunicación periódica con el sistema. La persistencia de la información será satisfecha a través del acceso a un historial de ejercicio realizado y la identificación personal. Además, es necesario que haya conectividad entre el dispositivo ECG (Vital Jacket) y el servidor web. Una vez transmitida la información al servidor, ésta será enviada a una base de datos, la cual también será necesaria.

ANÁLISIS DE MERCADO

Una vez iniciado el proyecto VitalGym sentimos la necesidad de saber si existía un lugar en el mercado para nuestro producto y de compararlo con otras alternativas ya existentes. Con ese propósito, entrevistamos a los responsables de varios gimnasios de la ciudad de Aveiro obteniendo los siguientes resultados y conclusiones.

El primero de ellos, *Knock-Out Gymnasium*, tenía un sistema de monitorización basado en una banda o cinta de monitorización del ritmo cardíaco. El ECG de los atletas era medido en tiempo real y los resultados eran anotados en papel, dentro de la ficha de cada atleta. Al no ser los resultados digitales, los atletas no tenían la posibilidad de ver su progreso. La saturación de oxígeno tampoco era medida, pero pretendían instalar un sistema para medir este parámetro. Este gimnasio se ofreció como campo de pruebas para que realizásemos las pruebas oportunas con nuestro producto y nos informaron de que estarían dispuestos a adquirir un sistema como VitalGym, especialmente si incluía monitorización de la saturación de oxígeno.

El siguiente gimnasio visitado fue el *Sports Pavilion Galitos*, donde las modalidades de entrenamiento eran estándar y, por lo tanto, no tenían un sistema de monitorización de la condición física y tampoco deseaban tenerlo. Esto excluía a VitalGym de este lugar.

Al igual que el *Knock-Out*, el gimnasio *Companhia de Fitness* tenía un sistema de monitorización basado en una cinta y los resultados de las sesiones de chequeo no eran digitales. Por lo que mostraron interés en la posibilidad del almacenamiento digital de dichas sesiones y en la medición de la saturación de oxígeno.

El *Gym Tónico Gymnasium* difería de los anteriores en la existencia de un sistema de almacenamiento digital y además los atletas podían ver los resultados de sus sesiones de chequeo. Sin embargo, no tenían una web donde los atletas pudieran ver su progreso ya que esto no era considerado como una característica imprescindible. Este gimnasio ya había experimentado con Vital Jacket, pero seguían prefiriendo la cinta o banda a no ser que el Vital Jacket fuera capaz de medir más parámetros.

El último gimnasio visitado fue el *Holmes Place*, que disponía monitorización a través de cinta, almacenamiento digital de resultados y carecía de sistema de medición de saturación de oxígeno. En general, estaban satisfechos con su sistema actual, pero admitieron que si Vital Gym llegaba a ser lanzado de forma estable estarían dispuestos a reunirse con nosotros para discutir la posibilidad de comenzar a usar un sistema de monitorización basado en Vital Jacket.

Conclusiones

Mientras que Vital Jacket cuesta aproximadamente 532€, un transmisor Polar T31 ronda los 29€. Es una gran diferencia de precio que puede conducir a los propietarios de los gimnasios a elegir un sistema de monitorización basado en una cinta, a pesar de que la precisión de Vital Jacket sea mucho mayor que la de una cinta o banda. Además, un Vital Jacket es mucho más cómodo a la hora de llevarlo. Por tanto, es evidente que un sistema de monitorización basado en Vital Jacket aún no es viable para la mayoría de los gimnasios, aunque sí lo es para los gimnasios de gama alta que pueden permitirse adquirir un sistema más caro pero a la vez más preciso. Sin embargo, si Vital Jacket pudiera medir la saturación de oxígeno, esto trasladaría nuestro proyecto a otro nivel completamente diferente.

Finalmente, de las entrevistas efectuadas también concluimos que nuestro sistema no debería realizar ninguna suposición acerca de la forma física de los atletas sino mejor otorgar dicha tarea al *Personal Trainer*.

Fuentes

- <http://www.optima-life.com/vitaljacket/howtoorder.html>
- http://www.amazon.co.uk/Polar-920130-T31-Uncoded-Transmitter/dp/B000N4N4UG/ref=sr_1_2?ie=UTF8&s=drugstore&qid=1300588377&sr=8-2

REQUISITOS DEL SISTEMA

Aquí se muestran las características que deberá incluir Vital Gym como sistema, el cual deberá proporcionar información precisa, exacta y coherente acerca de la resistencia física y eficiencia del atleta. El informe se centra en los requisitos funcionales de todo el sistema que serán divididos en las siguientes cualidades: usabilidad, fiabilidad, rendimiento y soporte.

Los requisitos de **usabilidad** incluirán aquellos basados en factores humanos y asuntos de la interfaz de usuario tales como la accesibilidad, la estética de la interfaz y la consistencia dentro de la interfaz de usuario. El sistema de Vital Gym tiene que ser fácil de usar y accesible para gente de todas las edades. La información referente al entrenamiento, la programación del mismo y el contacto con el entrenador será proporcionada en la web. Los atletas serán también capaces de obtener información acerca de su progreso en forma de gráficos.

Los requisitos de **fiabilidad** incluirán aspectos tales como la disponibilidad, precisión, previsibilidad, frecuencia de fallos o la recuperación del sistema ante un fallo de suministro eléctrico. El sistema debe proveer información acerca del progreso del atleta de forma muy precisa. Se usará la información del ritmo cardíaco en tiempo real de modo que el entrenador será capaz de proponer el entrenamiento más específico para el atleta en cuestión.

Los requisitos de **rendimiento** conciernen a rasgos del sistema tales como la eficiencia de la información a través del sistema, el tiempo de respuesta del sistema y el uso de los recursos. El flujo de información será enviado desde el chaleco (Vital Jacket) al servidor web mediante TCP/IP. Entonces toda la información será almacenada en una base de datos, la cual proporcionará acceso permanente al *Personal Trainer*.

Los requisitos de **soporte** deberán incluir aquellos requisitos tales como la compatibilidad y las opciones de prueba, ajuste, mantenimiento, configuración, instalación, escalado y localización del sistema.

Debemos indicar que muchos de los rasgos del sistema fueron determinados durante la fase de implementación del mismo, ya que es obvio que no era posible obtener toda la información durante la primera etapa de planificación y creación del sistema.

CASOS DE USO

En esta sección se muestran los *Casos de Uso* más relevantes de VitalGym, donde podremos ver las múltiples acciones que los actores del sistema pueden llevar a cabo.

Casos de Uso del Usuario del Sistema y el Administrador

Los actores son el **Usuario del Sistema**, representado por cualquier usuario genérico que pueda visitar la web; y el **Administrador**, que es la persona que representa al gimnasio.

La acción básica que podrá realizar el Usuario del Sistema será la de ‘login’, mientras que el Administrador será capaz de ‘añadir o eliminar Personal Trainers’, ‘ver la lista de Atletas y Personal Trainers’, ‘cambiar el Personal Trainer de un Atleta’ y por último, ‘cambiar datos personales’. Todas estas acciones son analizadas con mayor detalle en el *Anexo* (pág. xiv).

Diagrama de Casos de Uso

Aquí uno de los actores es representado por el entrenador o **Personal Trainer**, que se encargará de ‘añadir Atletas al sistema’ y gestionarlos. Tendrá Atletas bajo su supervisión y será responsable de ‘añadir sesiones de chequeo’ y ‘confeccionar programas de entrenamiento’, así como ‘modificar o eliminar’ dichos programas.

El otro actor que entrará en escena será, obviamente, el **Atleta**. Éste podrá ‘visualizar los resultados de sus sesiones de chequeo’, ‘comprobar su programa de entrenamiento’, ‘leer las anotaciones de su Personal Trainer’ y también ‘modificar sus datos personales’.

Como se ha indicado anteriormente, es recomendable leer el *Anexo* para obtener una información más detallada acerca de esta sección y ver con más claridad todos los *Casos de Uso* (Use Cases) a través de los diagramas que se muestran para tal propósito.

CASOS DE PRUEBA

A continuación se describen algunas pruebas realizadas sobre algunas de las acciones mostradas anteriormente para comprobar el correcto funcionamiento del sistema.

- **Login:** Para acceder a la web los usuarios deberán validar sus credenciales y para ello es necesario comprobar el nombre de usuario y la contraseña correspondiente, así como comprobar que la conexión a la base de datos se encuentra activa.
- **Eliminar Atleta:** Se comprueba que el Atleta a ser eliminado existe, si el entrenador tiene permisos para realizar dicha acción, y finalmente si el Atleta ha sido eliminado con éxito.
- **Crear Programa de Entrenamiento:** El Personal Trainer define una serie de ejercicios a realizar por el Atleta, basándose en los resultados de la sesión de chequeo y la evolución del Atleta. Para ello, se realizan comprobaciones de la última sesión de chequeo, de los permisos del entrenador, la identidad del Atleta y por último, si el programa ha sido creado.
- **Enviar Señales Vitales a la Base de Datos Online:** Este caso es crucial para el proyecto. La comunicación entre el Vital Jacket y el sistema es indispensable ya que para el análisis de los datos necesitaremos el flujo de información enviada desde el chaleco a la base de datos, donde se elaborará un histórico. La comprobación a realizar será si la conexión wireless se encuentra activa o no.
- **Ver resultados de la sesión de chequeo:** El Atleta deberá ser capaz de visualizar detalladamente la información online relacionada con su rendimiento en la última sesión de chequeo efectuada. Se debe comprobar si, efectivamente, se ha realizado una sesión de chequeo para ver sus resultados.

En el *Anexo* se pueden ver más *Casos de Prueba* (Test Cases) a tener en cuenta, así como todos los anteriores, claramente desglosados.

DISEÑO Y ARQUITECTURA

La arquitectura de VitalGym es descrita aquí para mostrar cómo deben ser divididos los diferentes componentes del sistema. La información proporcionada por todos los componentes será extremadamente útil durante todo el proyecto debido a que nos da una visión de conjunto del sistema.

Diagrama de clases

La información que este diagrama (ver *Anexo*) provee es de gran importancia, ya que su análisis nos permite dividir todas las clases en diferentes paquetes. Esos paquetes son:

- Usuarios del Sistema: referido a todos los usuarios del sistema (Atletas, Personal Trainers y Administrador).
- Análisis Online: representa todo el análisis que el sistema realizará “online”.
- Monitor: utilizado para el procesamiento del flujo de ECG.
- Análisis Offline: representa todo el análisis que el sistema efectuará “offline”.

Cada paquete contiene información referente a la clase a la que corresponde, sus atributos y métodos y la descripción del propio paquete.

Diagrama de componentes

Desde el punto de vista de los componentes (ver Diagrama de componentes en el *Anexo*), podemos dividir el sistema en los siguientes componentes:

- Análisis Online.
- Análisis Offline.
- Vital Jacket.
- VitaGym Monitor.
- Interfaz de Usuario (UI).
- Base de Datos (DB).
- Persistencia.

DIAGRAMAS DE ACTIVIDAD

En los diagramas de actividad que se encuentran en el *Anexo* podemos encontrar toda la información relacionada a:

- Adición de un nuevo atleta al sistema.
- Modificación del programa de entrenamiento.
- Vista previa de sesiones de chequeo o programas de entrenamiento.

Estos diagramas son en realidad diagramas de flujo que nos muestran, paso a paso, el algoritmo completo para la ejecución de la actividad en cuestión.

BASE DE DATOS

En el **diagrama de clases** y su **modelo físico** asociado (ver *Anexo*) encontraremos todo lo necesario para construir la base de datos de VitalGym. Para ello, nos hemos basado en las siguientes clases:

- **Tipo de Usuario:** contiene una lista de todos los tipos de usuario. Interactúa únicamente con la clase *Usuario*.
- **Usuario:** es el usuario que está utilizando el sistema. Contiene todos los atributos de una persona (nombre, email, contraseña...). Interactúa con las clases *Atleta*, *Personal Trainer* y *Administrador*.
- **Atleta:** generalización de este tipo de usuario. Puede interactuar con las clases *Usuario*, *Sesión de Chequeo* y *Personal Trainer*.
- **Personal Trainer:** generalización de este tipo de usuario. Cuando es borrado, se pondrá una marca para indicarlo y para permitir saber quién añadió las notas de una sesión de chequeo. Interactúa con *Usuario*, *Administrador*, *Nota* y *Atleta*.
- **Administrador:** representa a un gimnasio específico, ya que en la base de datos habrá información de varios gimnasios. Interactúa con *Usuario*, *Personal Trainer* y *Actividad*.
- **Sesión de Chequeo:** interactúa con las clases *Atleta*, *Nota*, *Informe de Sesión*, *Actividad de Sesión* y *Flujo ECG*.
- **Nota:** anotaciones sobre una sesión de chequeo específica. Es añadida por un entrenador. Interactúa con *Sesión de Chequeo* y *Personal Trainer*.
- **Informe de Sesión:** contiene el resultado final de una sesión de chequeo. Interactúa solamente con la clase *Sesión de Chequeo*.
- **Actividad de Sesión:** actividad realizada durante la sesión de chequeo. Las sesiones pueden tener varias. Interactúa con *Sesión de Chequeo* y *Actividad*.
- **Actividad:** es el tipo de actividad realizada. Interactúa con *Actividad de Sesión* y *Administrador*.
- **Flujo ECG:** información proporcionada por el Vital Jacket. Una sesión sólo puede tener un flujo cada vez. Interactúa con *Sesión de Chequeo*.

CÓDIGO Y DOCUMENTACIÓN DE DESARROLLO

Cada miembro del equipo ha comentado el código desarrollado en los formatos adecuados a dicho código:

Javadoc

La mayor parte del código ha sido desarrollado en Java y, por tanto, gran parte de la documentación se encuentra en formato javadoc.

En el enlace siguiente se puede ver casi toda la documentación generada en este caso:
<http://dl.dropbox.com/u/977251/javadoc/r224/site/apidocs/index.html>

Otro código

Evidentemente, no todo el código del proyecto está en Java y, por tanto, no hay javadoc para ese código. Los otros formatos de código desarrollados en este proyecto son:

- ❖ HTML/JavaScript
 - *live.html*
 - *offline.html*
- ❖ JSP
 - *live.jsp*
 - *offline_req.jsp*
 - *offline.jsp*
- ❖ JSF(XHTML)
 - *layout.xhtml*
 - *centerContent.xhtml*

ANEXO

VITALGYM

Real Time ECG Monitorization

PROJECTO EM ENGENHARIA INFORMÁTICA

Universidade de Aveiro, 2011

Nuno Silva
Marek Pastuszka
Frederico Honório
Rui Martins
José Mendes
Pedro Ramos

CONTENTS

INTRODUCTION	3
 DOCUMENTATION REQUIREMENTS AND ARCHITECTURE	4
Project Vision.....	4
Market Analysis	6
System Wide Requirements.....	8
Use Cases.....	9
Test Cases	14
Design and Architecture.....	25
Activity diagrams	29
Database Models.....	32
Layout Mock-Up	35
 CODE AND DEVELOPMENT DOCUMENTATION	43
Javadoc.....	43
Other code	43
HTML/JavaScript.....	43
JSP	44
JSF (XHTML)	45

Introduction

This product (VitalGym) intends to be a monitoring system of gym athlete's. As the physical shape of someone can be inferred from the heart response to physical exercise, the purpose of this system is to help gyms to establish an adequate training program within the individual needs of the athletes. With this system we hope the gyms won't need to establish a general training program to everyone (as it is nowadays) but instead that they establish adequate programs. This contributes to the athletes' satisfaction with the gym, which increases gym's reputation.

It is possible to watch a video presentation of our product at:

<http://www.youtube.com/watch?v=NdQXhn11bWo>

Documentation requirements and architecture

Project Vision

In this section is described a project vision. This means that it have information regarding the *Problem Statement*, *Product Position Statement*, *Stakeholder Descriptions* and finally a *Product Overview*.

1. Positioning

1.1. Problem Statement

The problem of	Bad interpretation of results obtained during the evolving of training sessions in gyms. Hard to understand physical limitations of athletes. The lack of a customized training program in gyms.
affects	Both the athletes and the credibility of gyms.
the impact of which is	Training programs which only please average population. Lack of continuous monitoring of improves in athlete's physical condition.
a successful solution would be	An application for monitoring heart rate and for process received data. This will measure the athlete's physical shape and provide bases to the creation of accurate customized training.

1.2. Product Position Statement

For	Gymnasiums
Who	Want to get better training plans without have a huge number of personal trainers in the work team.
The (product name)	VitalGym
That	It is proposed to improve the training plans providing information about the physical condition of the athlete to the personal trainer.
Our product	Will take place in market as a tool for gyms. As it will have an athlete's component where they can follow their progress, it will be the first to be gym-oriented and athlete-oriented at the same time.

3 - Stakeholder Descriptions

3.1. Stakeholder Summary

Name	Description	Responsibilities
Personal Trainer	Monitors the development of the athlete and plan his training.	The PT should adjust the athlete's training plan with the information provided by the system.
Athlete	Is the person who works out in the gym.	He is monitored by the system. He can also check his own progress and notes from the PT.
Administrator	The Administrator is the person who represents the gymnasium.	He can manage personnel (Personal Trainers) and change Athletes Personal Trainer. He is responsible for adding PT's to the system.

4 - Product Overview

4.1. Needs and Features

Need	Priority	Features
ECG monitoring	High	-ECG data source -Communicate periodically with the system
Information persistence	Average	-Access to the historic of exercises done -Personal identification

5 - Other Product Requirements

- It is necessary to have connectivity between the ECG device and the web server.
- After the ECG device transmits information to the server, this information will be send to a database. So a database connection is required.
- Vital Jacket is a must.

Market Analysis

Following the project VitalGym we felt the need to observe if there's a place in the market for our product and compare it with existent market alternatives. For that purpose, we interviewed a series of gymnasiums in Aveiro. The results of those interviews as well as conclusions are described in this report.

1. Gymnasiums

1.1. *Knock-Out Gymnasium*

Knock-out gym has a monitoring system. However it uses a heart-rate monitoring bandage. The electrocardiogram of the athletes is measured in real-time and is not stored in any kind of digital media. The real-time results are written down in paper and stored along with the athletes files. As the results are not digital, there is no possibility of the athletes see their results and consequently, their progress. Oxygen saturation is not measured but they plan to install a system that also measures this parameter.

p. It was offered any kind of assistance to our system including the possibility of field tests in the gym. The gym has already participated in an Aveiro University project and would be glad to be of assistance to VitalGym.

p. The gym also claimed that they would like to have a system like VitalGym especially if the monitoring features oxygen saturation monitoring.

1.2. *Sports Pavilion Galitos*

To observe if there is need of a system like VitalGym outside gymnasiums, we headed to sports pavilion Galitos. As the trainings for modalities are standard, they do not have a physical condition monitoring system nor do they would like to have. This excludes VitalGym of sports pavilions.

1.3. *Gymnasium Companhia de Fitness*

Like Knock-Out, Companhia de Fitness has a monitoring system based in a monitoring bandage and the check-up session results are not digital. They demonstrated interest in the possibility of digital storage of the check-up sessions and in the possibility of measuring the oxygen saturation.

1.4. *Gim Tónico Gymnasium*

Just like Knock-Out and Companhia de Fitness, Gim Tónico has a monitoring system based in a monitoring bandage. However, this gym has digital storage of the check-up session results. The results can also be viewed by athletes. They don't have a website where athletes can view their progress but they don't consider it to be a "must have" feature. This gym has already experimented Vital Jacket but they prefer a monitoring bandage. This could change if the Vital Jacket were able to measure more parameters.

1.5. *Holmes Place*

Holmes Place also has a monitoring system based in a monitoring bandage. This system also includes digital storage of the check-up session results. However it does not feature measurement of oxygen saturation. In general they are satisfied with the current system but if Vital Gym reaches a stable release, they would like to set up a meeting to discuss the possibility of start using a monitoring system based in Vital Jacket.

2. Conclusions

While Vital Jacket costs approximately 532€ a polar T31 transmitter costs approximately 29€. That's a huge difference in price which may lead gymnasium owners to choose a monitoring system based in a monitoring bandage. However Vital Jacket accuracy is greater than the accuracy of a monitoring bandage. Also, a Vital Jacket is more comfortable to wear than a bandage. Still a monitoring system based in Vital Jacket is not viable for regular gyms. It should focus in high-end, expensive gyms who can afford to buy a more expensive but more accurate system. However if Vital Jacket could measure oxygen saturation it would be at a completely different level. Since that at the start of our project Vital Jacket is not able to measure oxygen saturation, we should focus our system in high-end gyms.

From the performed interviews we also conclude that our system should not make many assumptions about the athletes' physical shape but leave the assumptions to the personal trainer.

3. Sources

- <http://www.optima-life.com/vitaljacket/howtoorder.html>
- http://www.amazon.co.uk/Polar-920130-T31-Uncoded-Transmitter/dp/B000N4N4UG/ref=sr_1_2?ie=UTF8&s=drugstore&qid=1300588377&sr=8-2

System Wide Requirements

This section contains information concerning the VitalGym system. VitalGym system will provide precise, accurate and coherent information about athlete's physical endurance and efficiency. Report will be focused on system-wide functional requirements which will be divided into following qualities: usability, reliability, performance and supportability.

- Usability requirements will include requirements based on human factors and user-interface issues such as accessibility, interface aesthetics, and consistency within the user interface.
- Reliability requirements will include aspects such as availability, accuracy, predictability, frequency of failure or recoverability of the system from shut-down failure.
- Performance requirements will concern traits of the system such as throughput of information through the system, system response time and resource usage.
- Supportability requirements will include requirements such as compatibility and the abilities to test, adapt, maintain, configure, install, scale, and localize the system.

It is obvious that in the first stage of planning and creating system requirements we can't provide all information due to the fact that many system's traits will be determined during implementation phase.

System Qualities

1. Usability

The VitalGym system has to be easy in use and accessible for people of all ages. The information about training, the schedule of trainings and the contact form with the personal trainer will be provided on the website. Athlete will be also able to get the information about the physical progress in the form of a graph.

2. Reliability

The system has to provide the information about athlete's progress so accuracy is one of the most important traits of it. The system will use the real-time information about athlete's heart rate so that personal trainer will be able to propose the most specific physical training to the athlete.

3. Performance

Information flow will be sent from vest to the server by TCP/IP. Then all the information will be stored in a database to provide permanent access for the personal trainer.

4. Supportability

There are no supportability requirements at this phase.

Use Cases

VitalGym most relevant *Use Cases* are described in the section. We can see the many actions that the actors can take.

1. System User and Administrator Use Cases

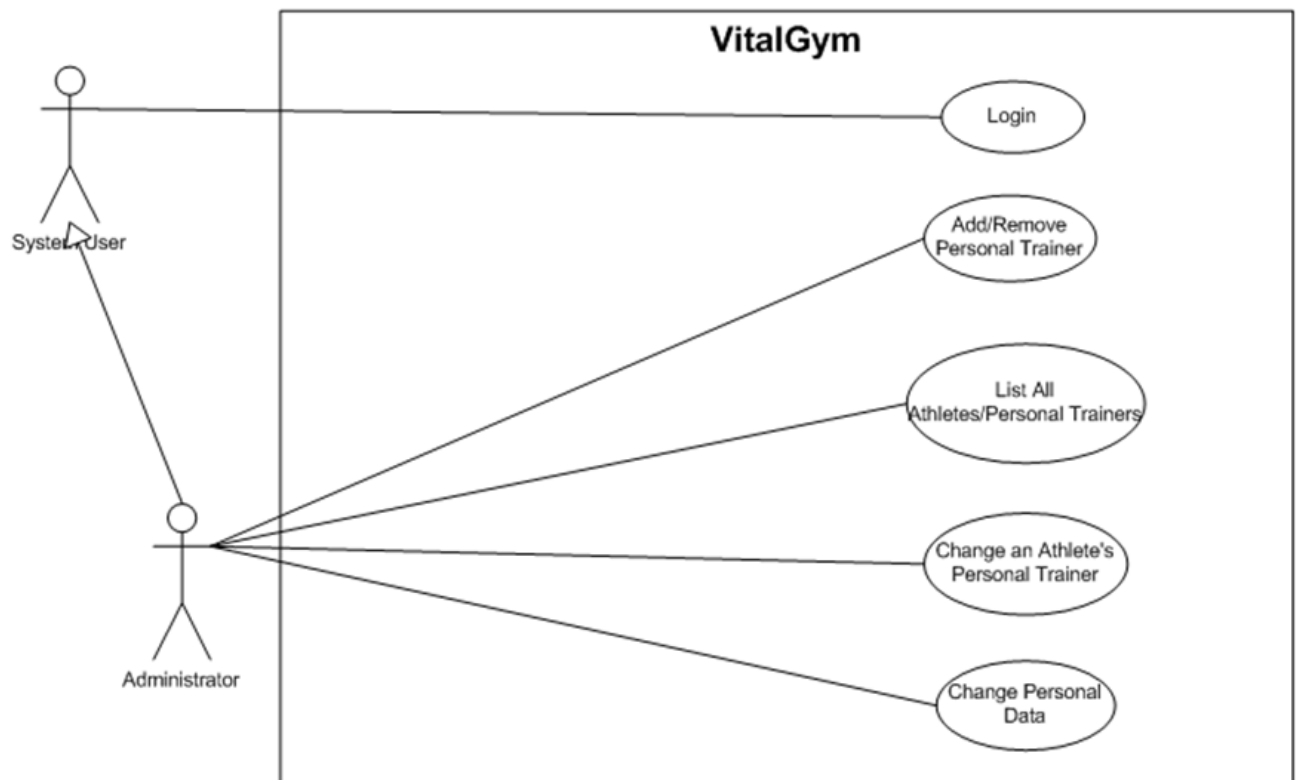


Diagram 1. Possible actions for System User and Administrator.

1.1. Actors

- **System User**
This actor represents a generic user. Anyone can visit the website. Such persons are System Users.
- **Administrator**
An Administrator is the person who represents the gymnasium.

1.2. Individual Use-Cases

- **Login**
A System User can login as a Personal Trainer, Administrator or as an Athlete, changing his role to Personal Trainer, Administrator or Athlete.
- **Add/Remove Personal Trainer**
An Administrator can add or remove Personal Trainers. It has taken the choice of this method for adding Personal Trainers to the system because, this way, there can only exist Personal Trainers associated with an Administrator and, therefore, a gymnasium.
- **List all Athletes/Personal Trainers**
An Administrator can view all the Athletes and Personal Trainers. This is needed for selection of an individual.
- **Change an Athlete's Personal Trainer**
An Administrator can change an Athlete's Personal Trainer. This comes in handy when a Personal Trainer leaves the gymnasium.
- **Change Personal Data**
As any other specialized Actor, an Administrator can change his personal data.

2. Use Case Diagram

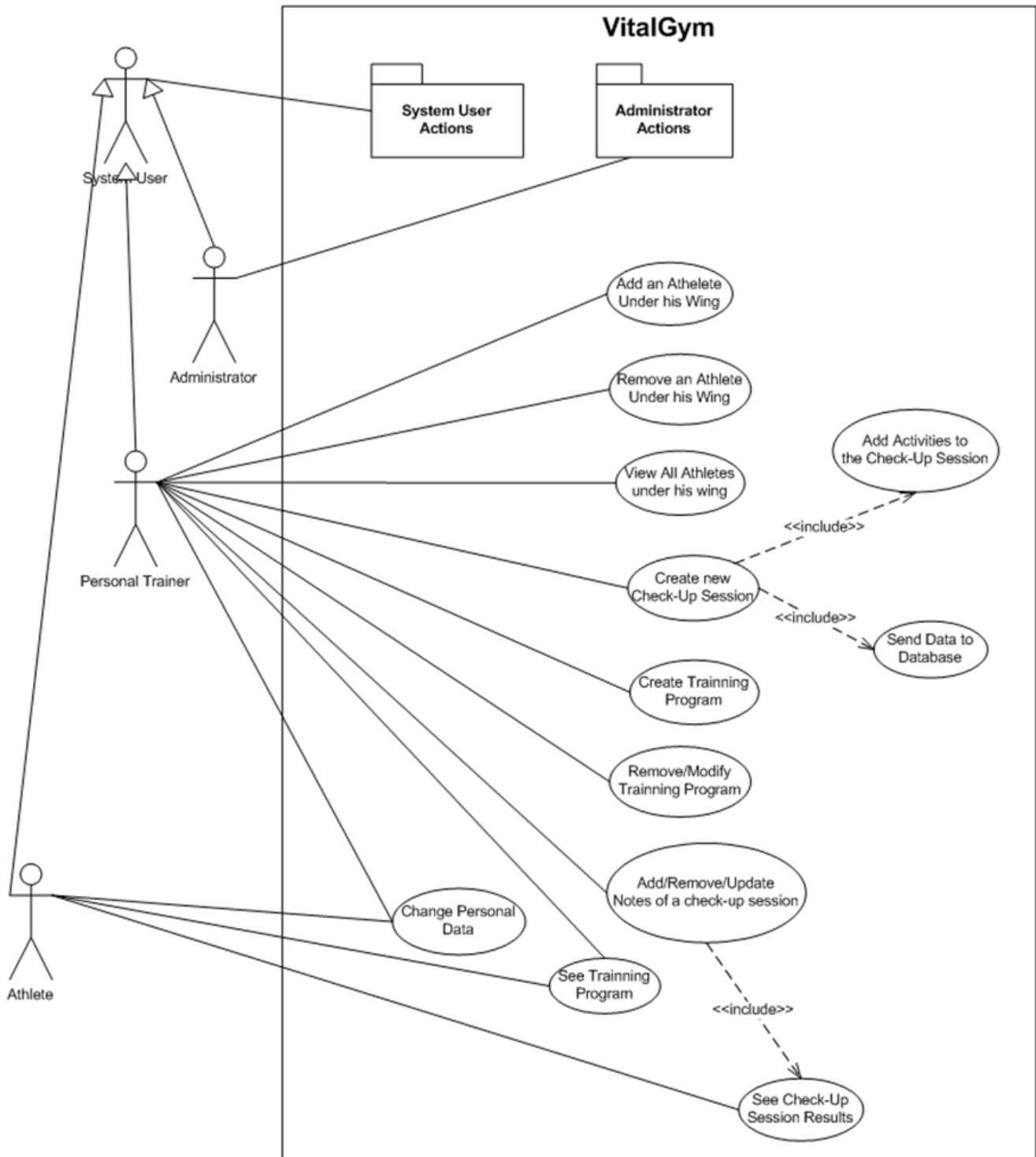


Diagram 2. Use Case Diagram of VitalGym with focus in the Personal Trainer and Athlete use cases.

2.1. Packages

Both System User Actions and Administrator Actions are packages who represent the use cases of Diagram 1.

2.2. Actors

- **Personal Trainer**

The personal trainer is the person who will add Athletes to the system and manage them. A Personal Trainer has Athletes under his wing and is responsible for adding check-up sessions and set up training plans as well as managing them.

- **Athlete**

An Athlete can see his own check-up session results, his training program as well as notes from the Personal Trainer responsible for him.

2.3. Individual Use-Cases

- **Add an Athlete under His Wing**

A Personal Trainer can add Athletes to the system. The Personal Trainer will be responsible for the added Athlete as well as managing their results.

- **Remove an Athlete under His Wing**

A Personal Trainer can remove an Athlete that is currently under his wing. This applies in cases where the Athlete has left the gym. To remove an Athlete, the Athlete must have been previously added to the system. A Personal Trainer cannot remove an Athlete that is under another Personal Trainer wing.

- **View All Athletes under his wing**

A Personal Trainer can see all his current Athletes. This is necessary to select a specific Athlete from all available.

- **Create New Check-Up Session**

A Personal Trainer creates Check-Up Sessions of an Athlete. This is essential to monitor the physical condition of the Athlete as well as recording progress associated with the training program. This only applies if the Athlete is under the Personal Trainer wing. The results of a Check-Up session will be transmitted to an online database.

- **Send Data to Database**

This use case represents the action of sending the results of a Check-Up session to the database.

- **Add activities to the Check-Up Session**

A Check-Up Session has activities. These activities are selected by the Personal Trainer.

- **Create Training Program**
A Personal Trainer can associate a training program with an Athlete with the purpose of improving the Athlete's physical shape. Obviously this only applies to Athletes under the Personal Trainer wing. An Athlete can only have zero or one training program associated with him.
- **Remove/Modify Training Program**
As the Athletes physical shape will improve over time, the Personal Trainer will feel the need to adjust the training program. If a great change is needed, it's easier to remove the previously established training program and create a new one. This only applies to previously create training programs.
- **Add/Remove/Update Notes of a Check-Up Session**
A Personal Trainer can create and manage notes of an Athlete's Check-Up session. More than one note of a Check-Up session can be created. The removal and update of notes only applies to previously added notes. The notes can also be public (visible to the Athlete and to the Personal Trainer) or private (for Personal Trainer concern only). To update or remove a Check-Up note, a Personal Trainer can see it first.
- **Change Personal Data**
Both Athlete and Personal Trainer can change their personal data. This applies, for example, when one wants to change his password.
- **See Check-Up Session Results**
An Athlete can access the website and see his Check-Up results online. That way, the Athlete can see for himself how his condition is improving. The public Personal Trainer Check-Up notes must also be visible.
- **See Training Program**
As well as Check-Up results, an Athlete must be able to see his training program. Public Personal Trainer training notes must be available as well.

Test Cases

1. Login

In order to access to the website users will need to login in the website. To validate their credentials several tests need to be done. This use case will have the next test cases:

1.1 <<test case #1 >>

Test Case ID: 1

Test Case Name: Check user id/username and password

Description: The system will check if the user ID exists and if the presented password matches the one corresponding to the ID stored in the Database.

Pre-conditions:

- *Database connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*

Post-conditions:

- *If successful: User get logged in the system*
- *If fail: An unsuccessful message will be prompt*

Data require: User information stored in database

1.2 <<test case #2 >>

Test Case ID: 2

Test Case Name: Check Database connection

Description: In order to have information in the website there will be needed a database. A test to this connection is a must. In this test will be tested if the connection is active and if the database is reachable.

Pre-conditions:

- *Wireless connection (Test case id: 12)*

Post-conditions:

- *If successful: A message will prompt saying that there is full connection to the database*
- *If fail: A error message will prompt*

Data required: Database address

2. Remove an Athlete

The Remove an Athlete from PT Wing is also an important use case. In this use case we will need to test if the athlete that we want to remove exists, if the PT has permissions to take that action and finally if the athlete was successful removed from his wing. The following tests will be made:

2.1 <<test case 1 >>

Test Case ID: 3

Test Case Name: Check Athlete ID

Description: In order to take any action over a system user it is strictly necessary to check if that user exists. In this case we are going to test if a user exists via its ID (a random generated number or membership number). This action will be done over the database doing a search. The action can be done to find a personal trainer. This test case is a must since we cannot remove anything if it doesn't exist.

Pre-conditions:

- *Database connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*
- *Require PT Login (Test case id: 1)*
- *PT permissions (Test case id: 4)*

Post-conditions:

- *If Successful: The athlete ID will be returned and it will be possible to remove the athlete from PT wing*
- *If fail: It won't be possible to remove the athlete since he doesn't exist and a message will prompt*

Data required: Information from database

2.2 <<test case 2 >>

Test Case ID: 4

Test Case Name: Check PT permissions

Description: This test case intends to test if the PT has permissions to make a specific action. To take an action, over athletes, the PT must have them associated to his account. This means that a personal trainer can't take actions over other personal trainer athlete's.

Pre-conditions:

- *Database connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*
- *Require PT Login (Test case id: 1)*

Post-conditions:

- *If successful: The PT will be able to take the intended action*
- *If fail: A message will prompt saying that PT don't have permissions to take that action*

Data required: Information from database

2.3 <<test case 3 >>

Test Case ID: 5

Test Case Name: Check if Removed

Description: This will be the final test case to the remove athlete use case. In this test we will check if the athlete was properly removed from the PT wing. This removal is not from the system but only from PT wing. This test should be done when the previously tests have been successful.

Pre-conditions:

- *Data base connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*
- *Require PT Login (Test case id: 1)*
- *Require Valid Athlete id (Test case id: 3)*

Post-conditions:

- *If successful: The athlete will be removed from PT wing and the database will be updated*
- *If fail: The athlete will not be deleted and the database will stay intact*

Data required: Information from database

3. Create an Athlete

The VitalGym System would not be reliable if it didn't have system users. In order to have users in a system it is necessary to create them. The presented group of tests to this use case will show us if a user was correctly created and if he is able to use the full functionalities presented by the website.

3.1 <<test case 1 >>

Test Case ID: 6 Test Case Name: Check Athlete email

Description: The new gym user should provide his email so his account can be created. This test will aim for a search of the presented email in the database. This is a security action since the system will not support equal emails for two users.

Pre-conditions:

- *Database connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*

Post-conditions:

- *If successful: The account should be created and the athlete should receive an email with a link to set his password*
- *If fail: A warning message will prompt with information*

Data required: Email registers in database

3.2 <<test case 2 >>

Test Case ID: 4

Test Case Name: Check PT permissions

Already described on test case with id 4. (Remove an athlete – test case 3)

3.3 <<test case 3 >>

Test Case ID: 7

Test Case Name: Check if Created

Description: This will be the final test case to the create athlete use case. In this test we will check if the athlete was properly created in the system. This creation doesn't include an associated PT yet. The test should be done when the previously tests have been successful.

Pre-conditions:

- *Database connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*
- *Require PT Login (Test case id: 1)*

Post-conditions:

- *If successful: The account should be created and the athlete should receive an email with a link to set his password*
- *If fail: A warning message will prompt with information*

Data required: Information from database

4. Add an Athlete

Like the use case for removal, the Add an Athlete Under PT Wing use case is extremely important and also need a few test cases. Every athlete should have a PT and for this, the PT should add the athlete, in the website, to be under his supervision. The PT will be responsible for defining the training program. For this option there will be a few tests like if the athlete really exists so he can be added under PT wing and if the PT got the right permissions.

4.1 <<test case 1 >>

Test Case ID: 4

Test Case Name: Check PT permissions

Already described on test case with id 4. (Remove an athlete – test case 3)

4.2 <<test case 2 >>

Test Case ID: 3

Test Case Name: Check Athlete ID

Already described on test case with id 3. (Remove an athlete – test case 1)

4.3 <<test case 3 >>

Test Case ID: 8

Test Case Name: Check if Added

Description: In this test will be tested if the athlete was correctly added under a specific PT supervision. The database will be checked assuming that the pre-conditions were satisfied.

Pre-conditions:

- *Database connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*
- *Require PT Login (Test case id: 1)*
- *Require Valid Athlete id (Test case id: 3)*

Post-conditions:

- *If successful: The athlete will be associated to a PT and the database will be updated*
- *If fail: The athlete will not be associated to a PT and the database will stay intact*

Data required: Information from database

5. Create new Check-Up Session

The objective of this use case is to create a new check-up session for a starter athlete. There will be run several tests in order to get some information about the athlete physical condition.

5.1 <<test case 1 >>

Test Case ID: 4

Test Case Name: Check PT permissions

Already described on test case with id 4. (Remove an athlete – test case 3)

5.2 <<test case 2 >>

Test Case ID: 3

Test Case Name: Check Athlete ID

Already described on test case with id 3. (Remove an athlete – test case 1)

5.3 <<test case 3 >>

Test Case ID: 9

Test Case Name: Check Check-Up Creation

Description: This test will be focused in the creation of the check-up session. To achieve a positive result on this test the database must be consulted and should have changed in the way that the PT and the athlete agreed. This change is reflected in the day that both agreed (like a calendar in both profiles).

Pre-conditions:

- *Database connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*
- *Require PT Login (Test case id: 1)*
- *Require Valid Athlete id (Test case id: 3)*

Post-conditions:

- *If successful: The check-up session should be schedule and the database should be updated*
- *If fail: A warning message will prompt with information*

Data required: Information from database, including user and PT ids.

6. Create Training Program

The creation of a training program consists in defining a few exercises that the athlete should do along a few weeks, established by the personal trainer. The training program is based in the check-up session results as well as the evolution of the athlete.

6.1 <<test case 1 >>

Test Case ID: 10

Test Case Name: Check Check-Up session

Description: The goal of this test case is to check if any check-up session was made. A training program can only be given to an athlete after he performs a check-up session. If the athlete didn't do a check-up session the test will fail. As resume, the plan can only be given after the physical condition of the athlete has been determinate.

Pre-conditions:

- *Database connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*
- *Require PT Login (Test case id: 1)*

Post-conditions:

- *If successful: This means that a previously session was made and creation of the training program can proceed*
- *If fail: An error message will prompt*

Data required: Data from database with user and PT information

6.2 <<test case 2 >>

Test Case ID: 4

Test Case Name: Check PT permissions

Already described on test case with id 4. (Remove an athlete – test case 3)

6.3 <<test case 3 >>

Test Case ID: 3

Test Case Name: Check Athlete ID

Already described on test case with id 3. (Remove an athlete – test case 1)

6.4 <<test case 4 >>

Test Case ID: 11

Test Case Name: Check Plan Creation

Description: After the creation of a plan we should check if the plan was really created. This can be done by checking the database and the athlete profile. The profile should have an associated plan. If the plan was update to the correct one the test should pass.

Pre-conditions:

- *Database connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*
- *A login*

Post-conditions:

- *If successful: The program should be created and showed in the user profile. The database should also be updated with such information.*
- *If fail: A warning message will prompt with information*

Data required: User ID and Person Trainer ID

7. Send Vital Signs to Online Database

This use case is crucial for the project. The communication between Vital Jacket and the system is imperative and this use case represents it. For data analysis we will need the information from the Vital Jacket and the database so we can make a historic. An important flow of information is here represented.

7.1 <<test case 1 >>

Test Case ID: 12

Test Case Name: Check Wireless Connection

Description: In order to have and see information in the website there will be needed a Wireless/Wired connection. In this test the connection should be checked if it is active or not.

Pre-conditions:

- *Have an internet connection (ISP)*

Post-conditions:

- *If successful: Should be allowed to send information for database*
- *If fail: A message should prompt saying that there isn't any active or working connection*

Data required: ISP credentials

8. Remove/Modify Training Program

This use case represents the action of remove or modifies a training program by the PT. Those actions can be made when the athlete no longer wants to continue his workout or when the amount of exercise should growth.

8.1 <<test case 1 >>

Test Case ID: 4

Test Case Name: Check PT permissions

Already described on test case with id 4. (Remove an athlete – test case 3)

8.2 <<test case 2 >>

Test Case ID: 3

Test Case Name: Check Athlete ID

Already described on test case with id 3. (Remove an athlete – test case 1)

8.3 <<test case 3 >>

Test Case ID: 13

Test Case Name: Check if Plan Exists

Description: This test case should be used to check if an athlete already has an associated program. This can be done by finding the user ID and then checking is profile. If he doesn't have any program yet, should be schedule a check-up session.

Pre-conditions:

- *Database connection (Test case id: 2)*
- *WIFI connection (Test case id: 12)*
- *Require Valid Athlete id (Test case id: 3)*

Post-conditions:

- *If successful: The personal trainer should be able to modify or remove an existing program*
- *If fail: The athlete must have a training program*

Data required: User id, training program information

9. Add/Update Notes of a Check-Up Session

This use case represents the action of Add/Update Notes of a check-up session by the PT. Those notes represent some additional information that a PT wants to give to an athlete.

9.1 <<test case 1 >>

Test Case ID: 3

Test Case Name: Check Athlete ID

Already described on test case with id 3. (Remove an athlete – test case 1)

9.2 <<test case 2 >>

Test Case ID: 4

Test Case Name: Check PT permissions

Already described on test case with id 4. (Remove an athlete – test case 3)

10. Add/Update Notes of a Training Program

This use case represents the action of Add/Update Notes of a Training Program by the PT. Those notes represent some additional information that a PT wants to give to an athlete regarding the training program.

10.1 <<test case 1 >>

Test Case ID: 3

Test Case Name: Check Athlete ID

Already described on test case with id 3. (Remove an athlete – test case 1)

10.2 <<test case 2 >>

Test Case ID: 4

Test Case Name: Check PT permissions

Already described on test case with id 4. (Remove an athlete – test case 3)

11. See Check-Up Session Results

This action is done by the athlete. He should be able to see detailed information online, regarding his performance in the check-up session.

11.1 <<test case 1 >>

Test Case ID: 14

Test Case Name: Check previously Sessions

Description: This test case intends to test if there were any previously check-up session done. If there wasn't the athlete shouldn't be able to see any results since he haven't perform any training yet.

Pre-conditions:

- *Database connection (Test case id: 2)*
- *Wireless connection (Test case id: 12)*
- *Require Athlete Login*

Post-conditions:

- *If successful: Should be presented his results*
- *If fail: A error message should prompt*

Data required: User ID

12. See Training Program

This action is done by the athlete. He should be able to see detailed information online, regarding his performance in the check-up session.

12.1 <<test case 1 >>

Test Case ID: 13

Test Case Name: Check if Plan Exists

Already described on test case with id 13. (Remove/Modify Training Program – test case 3)

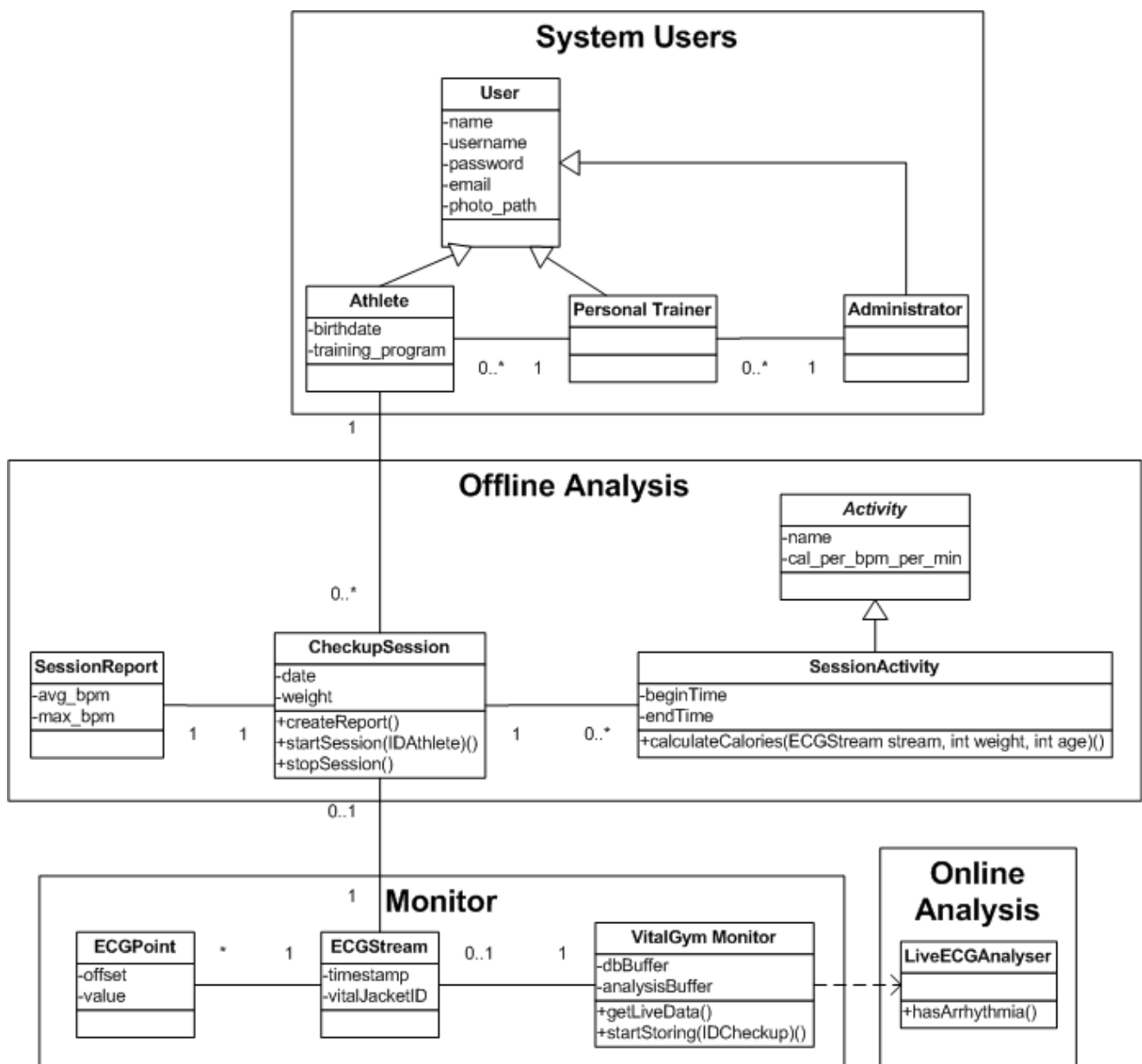
Design and Architecture

This section describes Vital Gym architecture and intends to be a guide on how the different system components should be divided. The information provided by all components will be extremely useful along the project since they give us an overall view of the system.

Domain model

In the domain model we define a class diagram and also a component diagram. Since these diagrams are in the *design and architecture* section, this means that the information provided by them is extremely important.

1. Class Diagram



As we can see by the analysis of the picture, we divide all classes in different packages. Those main packages are:

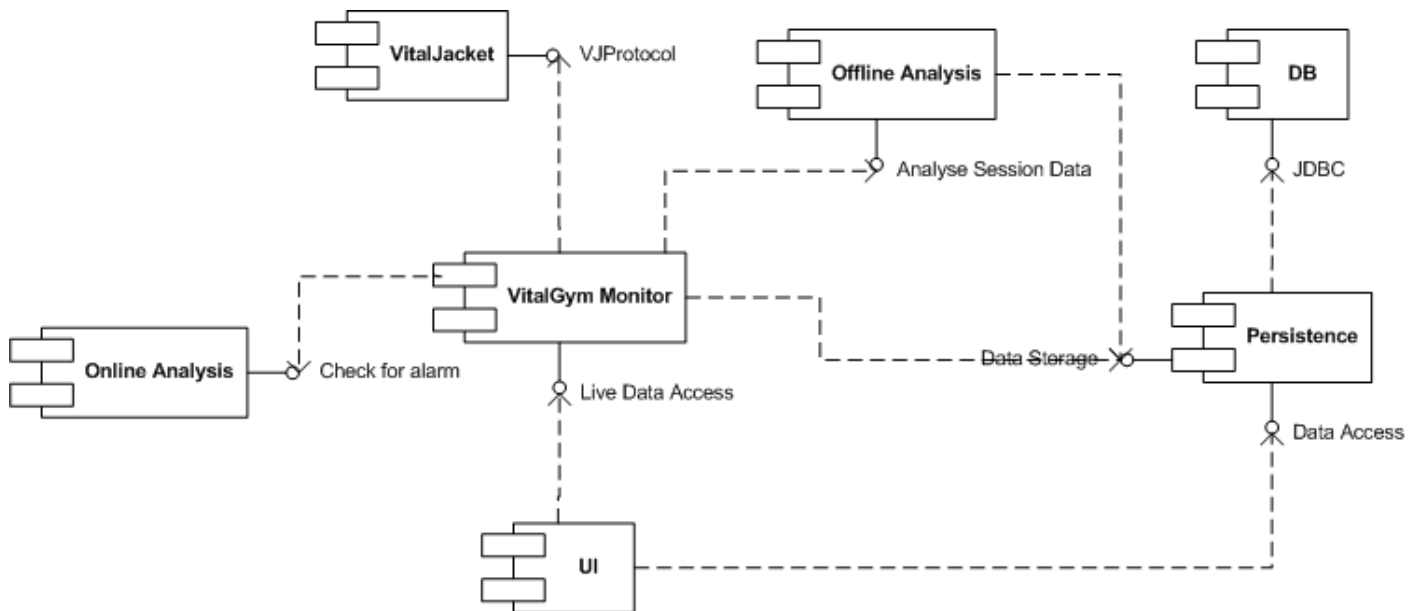
- System users - Concern with all system users (Athlete, PT and Administrator)
- Online Analysis - Represent all the analysis that the system will do " *online* "
- Monitor- Used for ECG stream processing
- Offline Analysis: - Represent all the analysis that the system will do " *offline* "

Each package contains the following information:

Package	Class	Attributes	Methods	Description
System Users	User	name	---	User name
System Users	User	username	---	Username used for login
System Users	User	password	---	Password used for login
System Users	User	email	---	User email, can be used for password recovery as well as to receive news/information
System Users	User	photo_path	---	User photograph to be displayed in the user profile
System Users	Athlete	birthdate	---	Athlete birthdate, it give us his age
System Users	Athlete	training_program	---	Provide the athlete actual training program
System Users	Personal Trainer	---	---	Inherits user attributes
System Users	Administrator	n/a	---	Inherits user attributes
Offline Analysis	SessionReport	avg_bpm	not yet defined	Give the average heart beats
Offline Analysis	SessionReport	max_bpm	not yet defined	Provide the maximum heart beats
Offline Analysis	CheckUpSession	Date	---	Date that was schedule for the checkup session
Offline Analysis	CheckUpSession	Weigth	---	Actual weigth of the athlete in the checkup session
Offline Analysis	CheckUpSession	---	createReport	Creates a detailed report of the result of the checkup session

Package	Class	Attributes	Methods	Description
Offline Analysis	CheckUpSession	---	startSession(IDAthlete)	Indicates that a checkup session has started (start the session)
Offline Analysis	CheckUpSession	---	stopSession	Indicates that a checkup session has finished (stop the session)
Offline Analysis	SessionActivity	beginTime	---	Give us the beginning time of an activity
Offline Analysis	SessionActivity	endTime	---	Give us the ending time of an activity
Offline Analysis	SessionActivity	---	calculateCalories(ECGStream stream, int weight, int age)	Function used to calculate the calories spent by minute
Offline Analysis	Activity	name	---	Name of a specific activity (Running, Weightlifting...)
Offline Analysis	Activity	cal_per_bpm_per_min	---	Calories per minute
Monitor	ECGPoint	offset	---	
Monitor	ECGPoint	value	---	
Monitor	ECGStream	timeStamp	---	Timestamp of the ECGStream received
Monitor	ECGStream	vitalJacketID	---	Provide the VitalJacketID
Monitor	VitalGymMonitor	dbBuffer	---	Buffer used to store/access data in/from database
Monitor	VitalGymMonitor	analysisBuffer	---	Buffer used to make possible a Live Analysis (For instance live graph)
Monitor	VitalGymMonitor	---	getLiveData	Get live data from the ECG
Monitor	VitalGymMonitor	---	startStoring(IDCheckup)	Start storing data in the database. This data is provided by a specific checkup session
OnLine Analysis	LiveECGAnalyser	---	hasArrhythmia	Verify if the athlete has any arrhythmia

2. Component Diagram

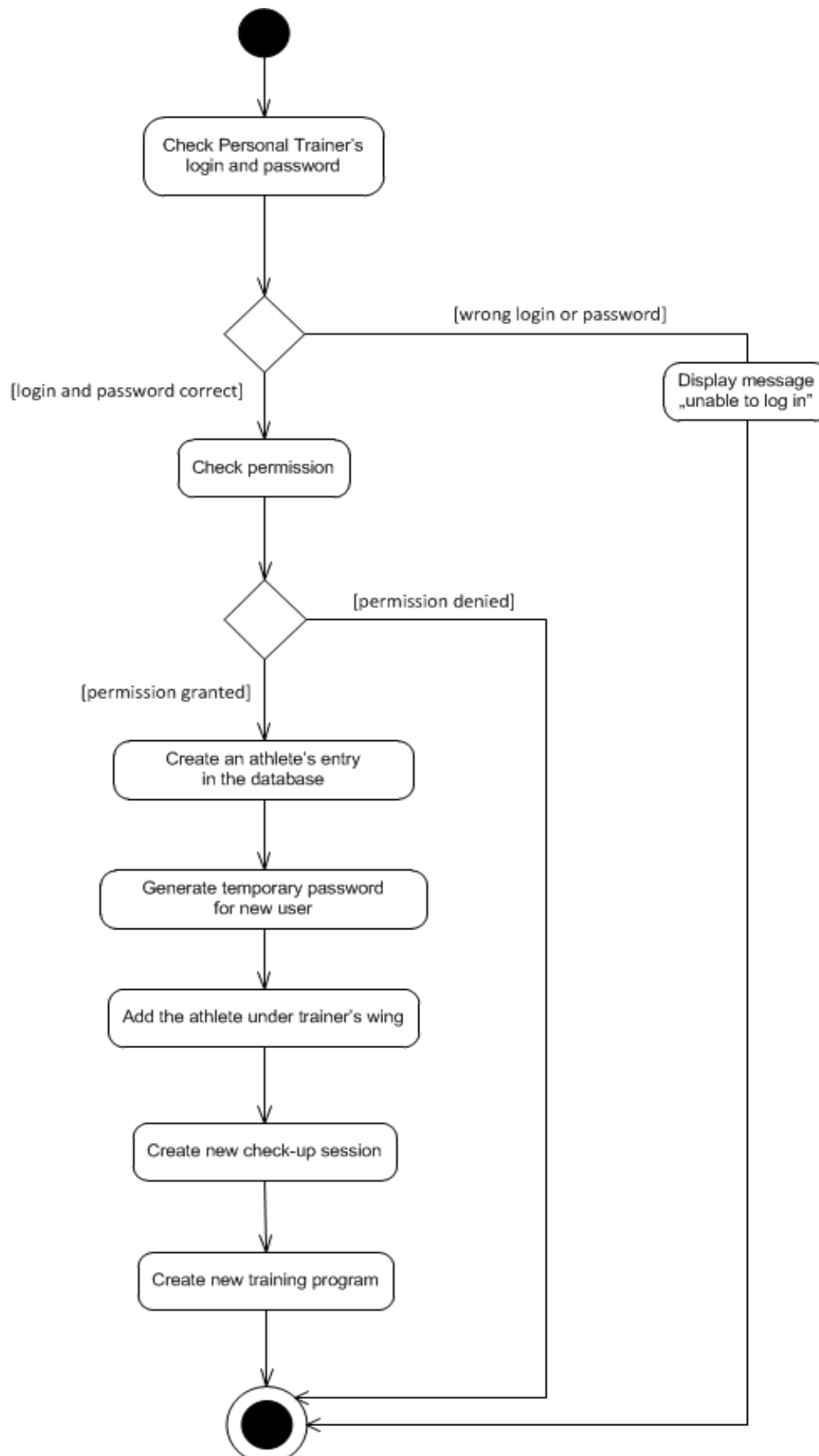


From the component point of view, we can see that we divide the system in the following components:

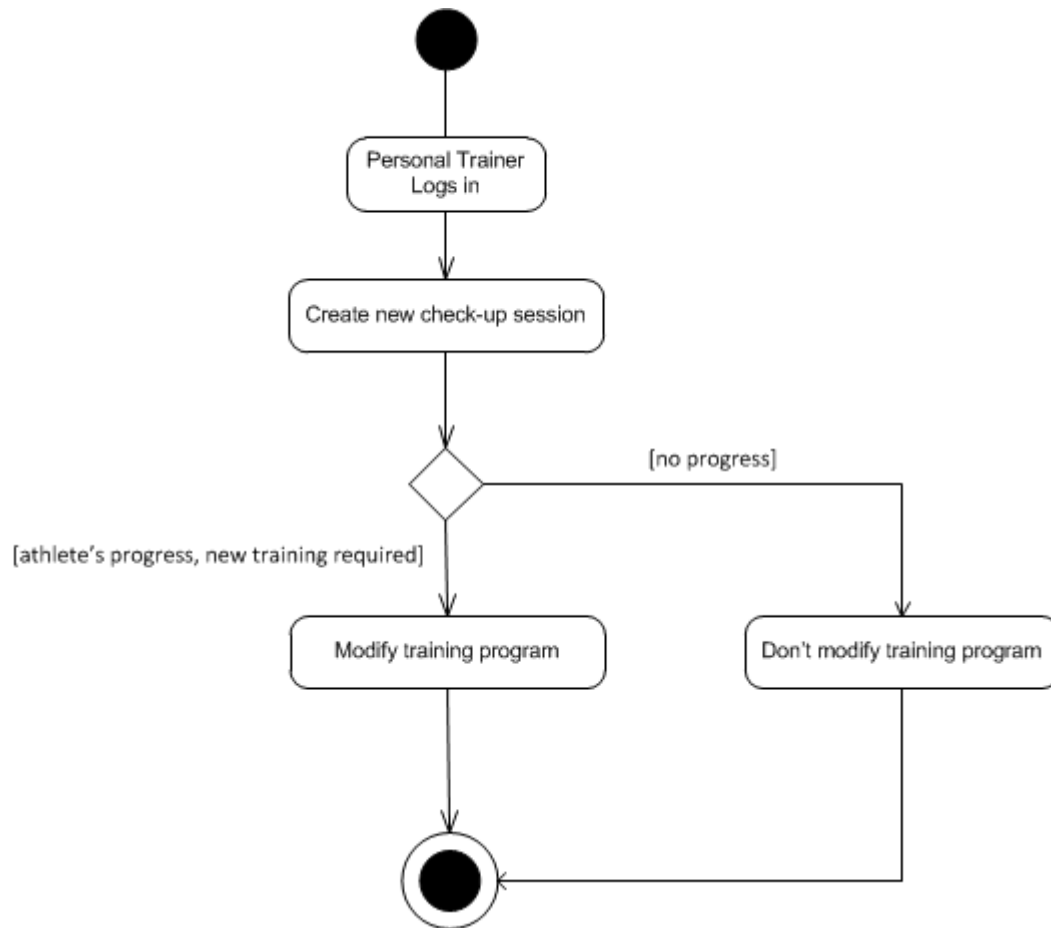
- Online Analysis
- Offline Analysis
- VitalJacket
- VitalGym Monitor
- UI
- DB
- Persistence

Activity diagrams

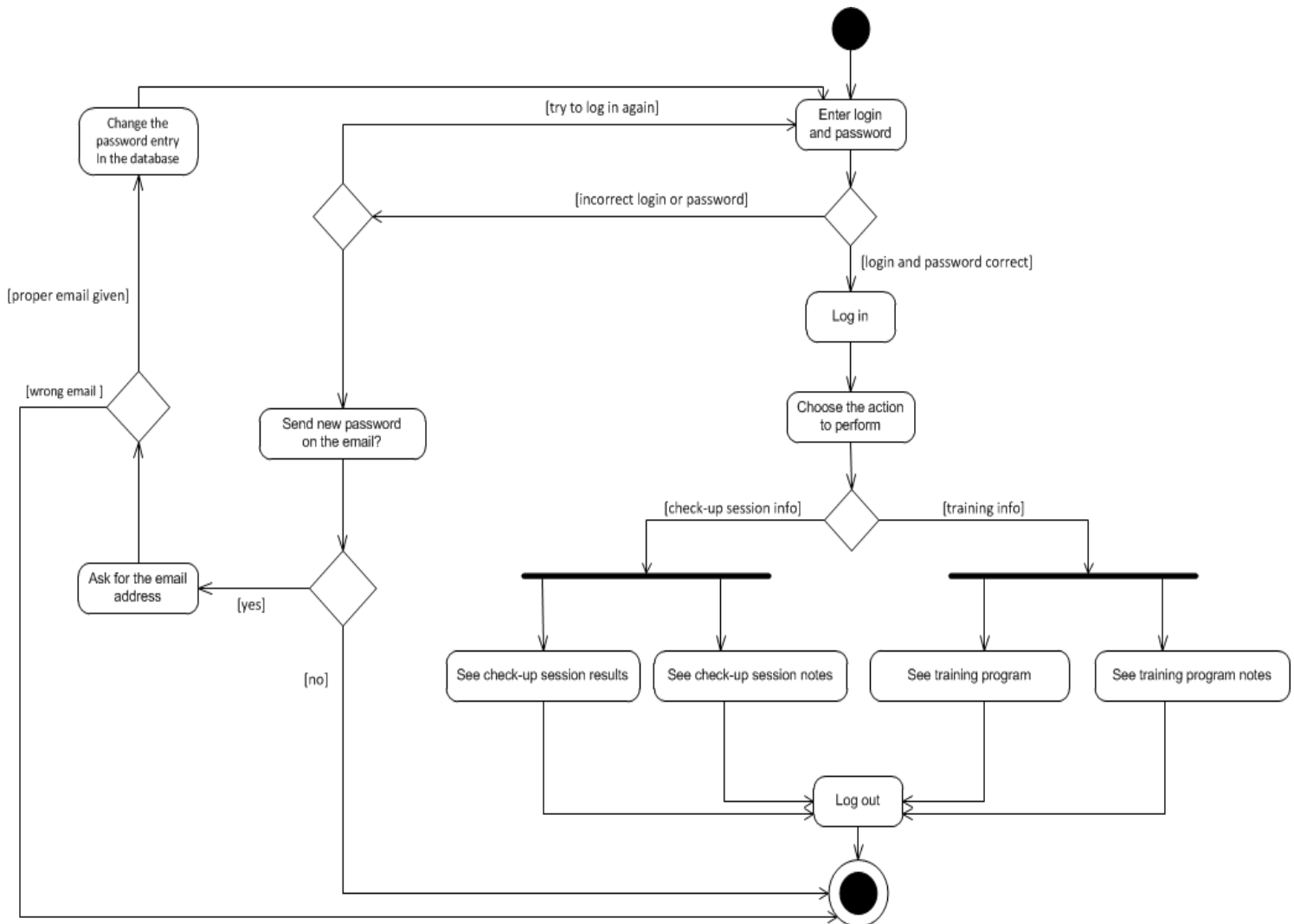
1. Activity concerning adding new athlete to the system



2. Activity concerning modifying training program



3. Activity concerning stats preview

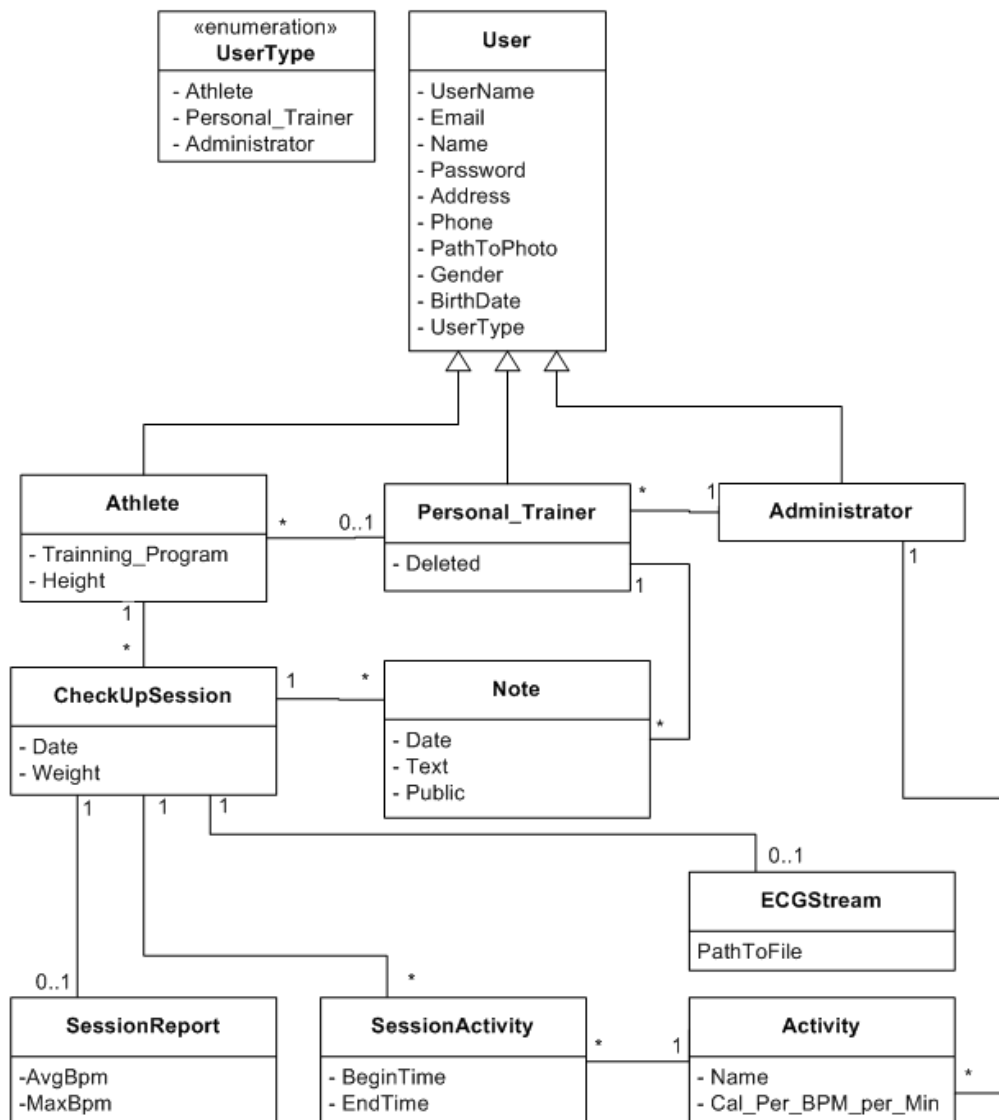


Database Models

This section contains all the necessary requirements in order to deploy a database. This means that this section contains the class diagram as well as the associated physical model.

1. Class Diagram

1.1. Diagram



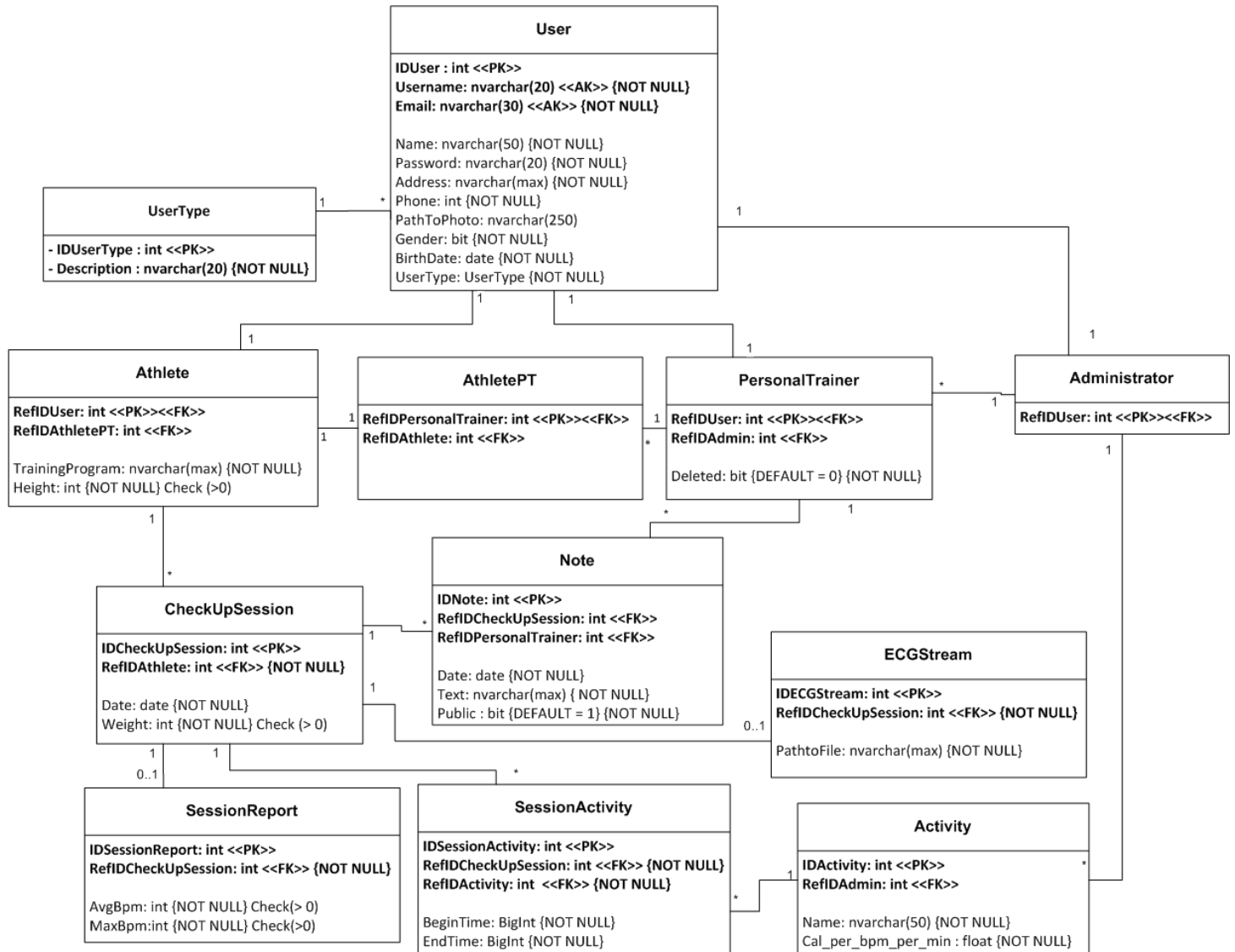
1.2 Documentation

For diagram support check this documentation:

Class Name	Description	Interactions
UserType	Enumeration containing all user types	User
User	User in the system. Contains all attributes of a person	Athlete - Personal_Trainer - Administrator
Athlete	Generalization of User class. It's a type of user who works out in the Gym	User - CheckUpSession - Personal_Trainer
Personal_Trainer	Generalization of User class. Is the athlete personal trainer. When deleted, a flag (deleted) will be set. This allows the Personal Trainer to be deleted but also provides a way to know who added the notes of a check-up session.	User - Administrator - Note - Athlete
Administrator	The administrator is used as the Gym. This means that the Administrator represents a specify gym	User - Personal_Trainer - Activity
CheckUpSession	CheckUpSession of an athlete. Have notes, a session report, an Athlete and various activities	Athlete - Note - SessionReport - SessionActivity - ECGStream
Note	Note of a specific check-up session. Is added by the personal trainer	CheckUpSession - Personal_Trainer
SessionReport	Contains the final result of the check up session.	CheckUpSession
SessionActivity	Activity done in the check up session. The sessions can have various	CheckUpSession - Activity
Activity	Have the type of activity	SessionActivity - Administrator
ECGStream	Stream provided by the Vital Jacket. One session can only have on stream at a time	CheckUpSession

2. Physical Model

2.1. Diagram



Layout Mock-Up

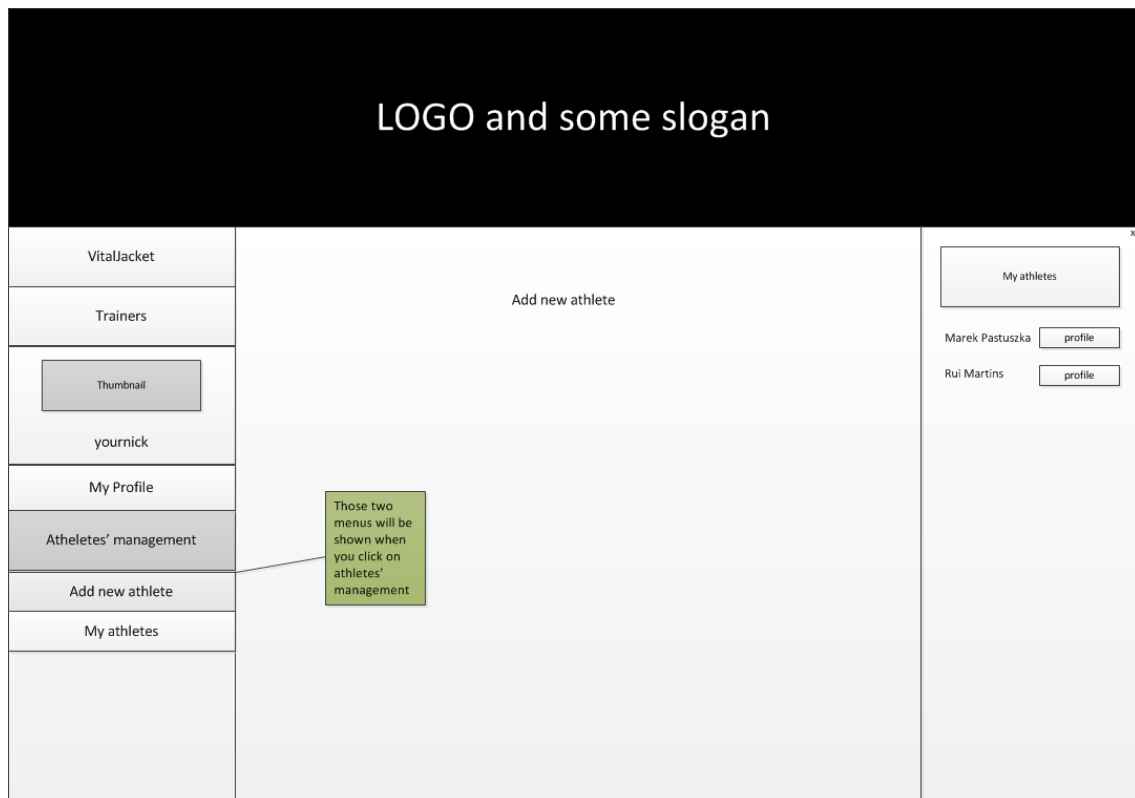
This section contains the information and the skeleton of the website that our team is working on right now.

We changed all prototypes to have the whole outlook on the functionality of the application that will fulfil all of the requirements and use cases. What's more we were trying to make the website as easy and efficient in use as it can be and what is important the mock-up was created after gaining the knowledge about existing widgets and elements that can be used in the creation of the website.

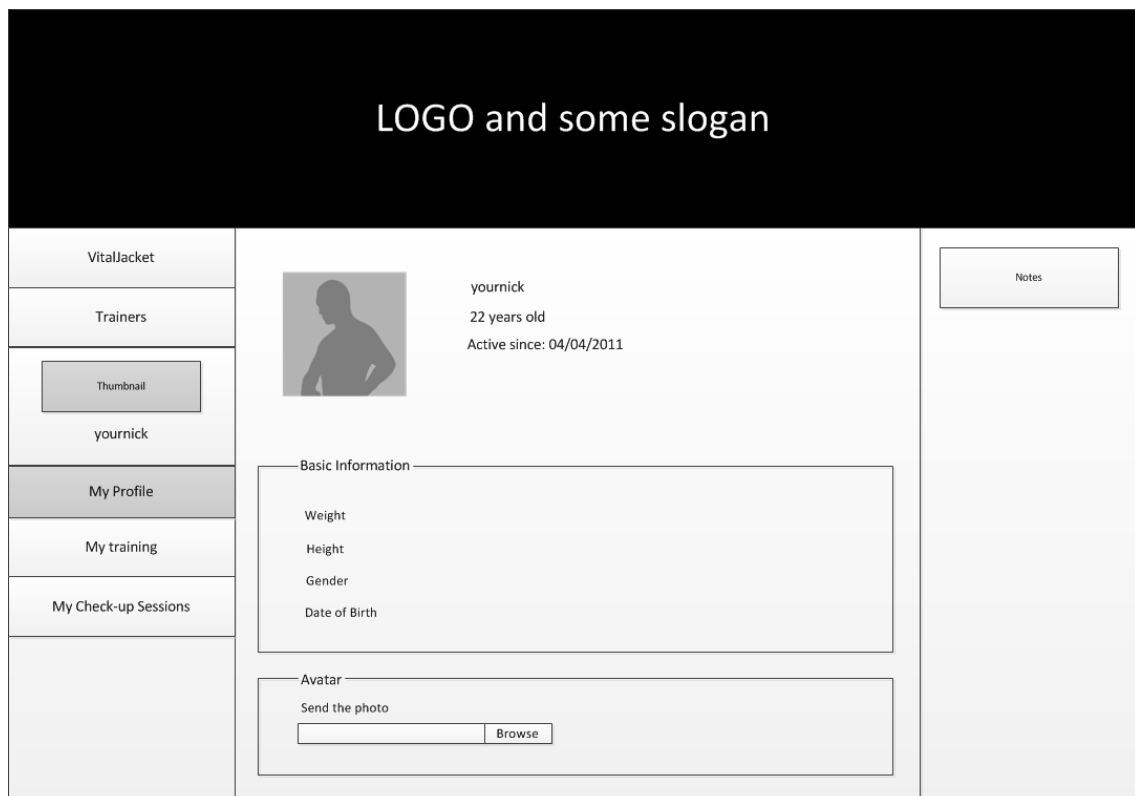
LOGO and some slogan

VitalJacket	<div>Last Check-Up Sessions:</div> <table><thead><tr><th>Title</th><th>Date</th><th>Notes</th></tr></thead><tbody><tr><td>Initial Session</td><td>30/04/2011</td><td></td></tr><tr><td>Progress one</td><td>30/05/2011</td><td></td></tr></tbody></table> <div>Click on the title to see the graph and the activity</div>	Title	Date	Notes	Initial Session	30/04/2011		Progress one	30/05/2011		Notes
Title		Date	Notes								
Initial Session		30/04/2011									
Progress one		30/05/2011									
Trainers											
<div>Thumbnail</div> <div>yournick</div>											
My training											
My Check-up Sessions											

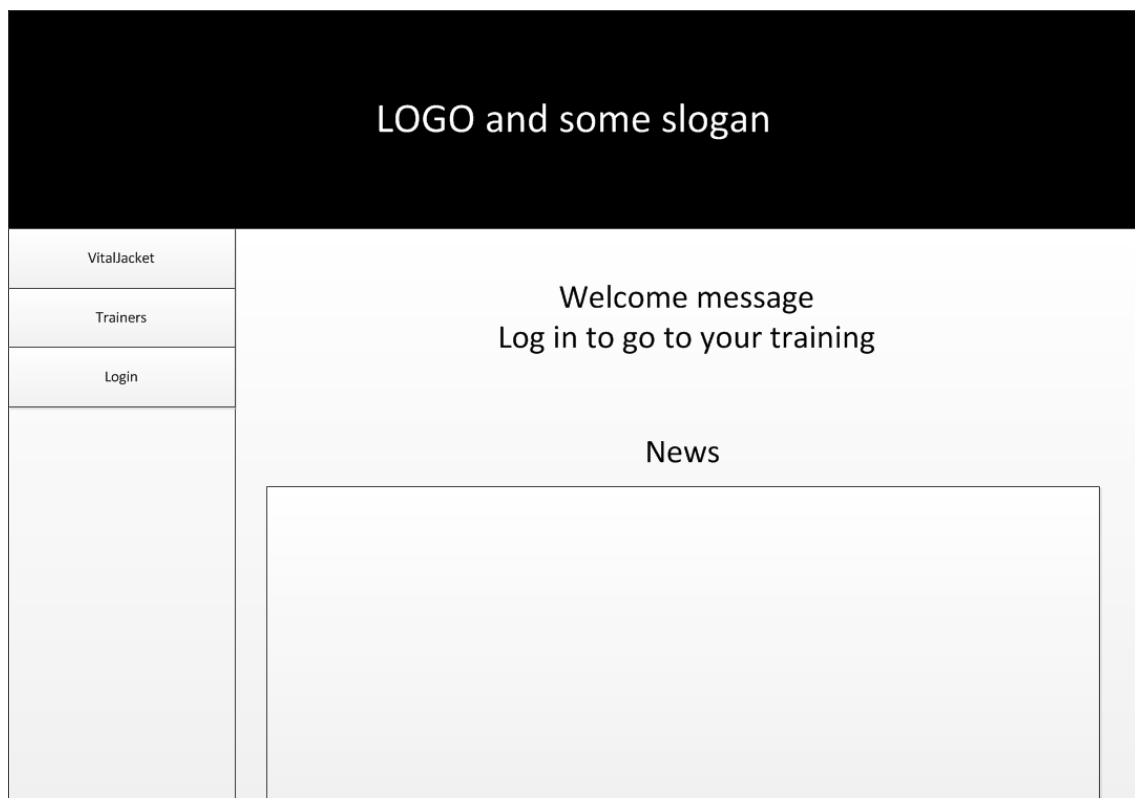
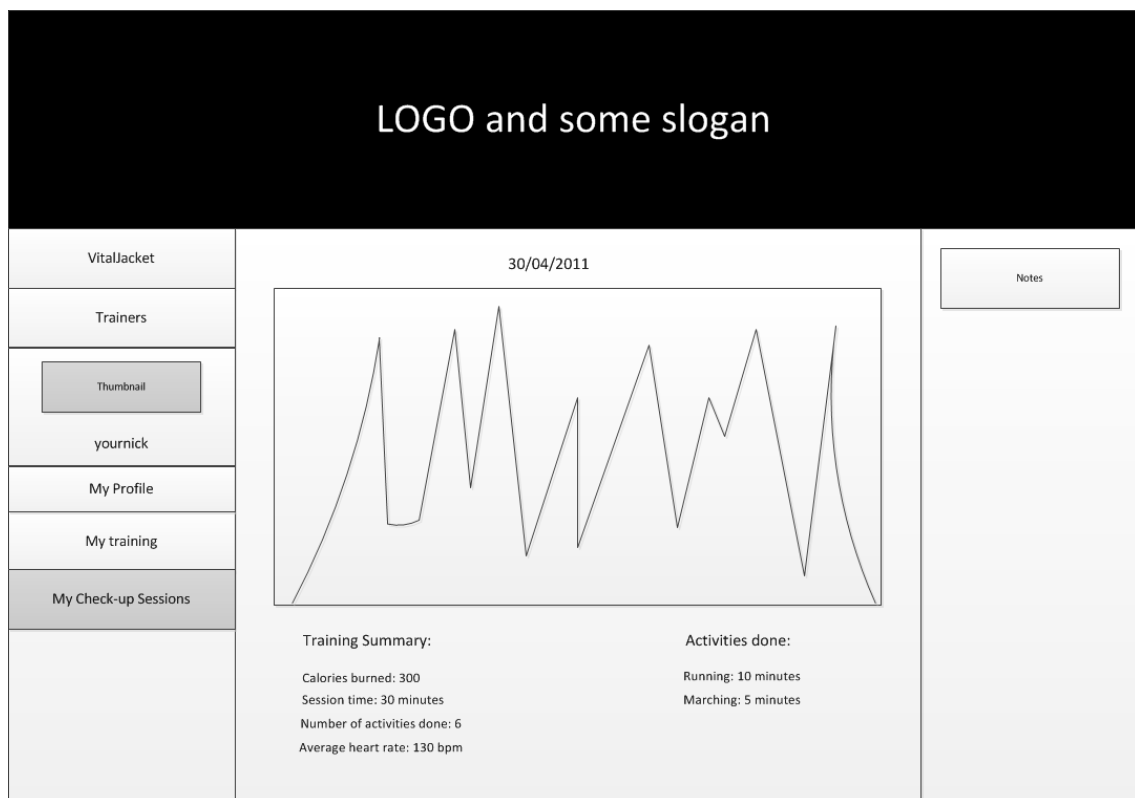
Check-Up sessions

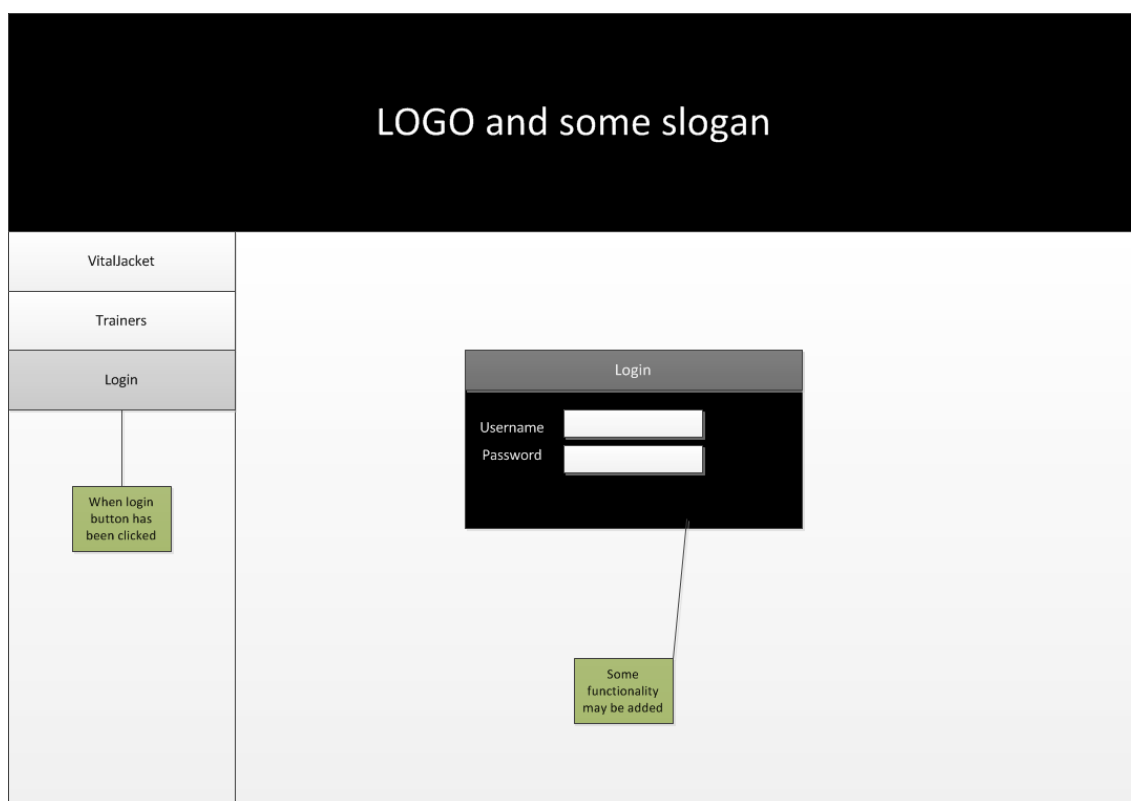


Add new Athlete

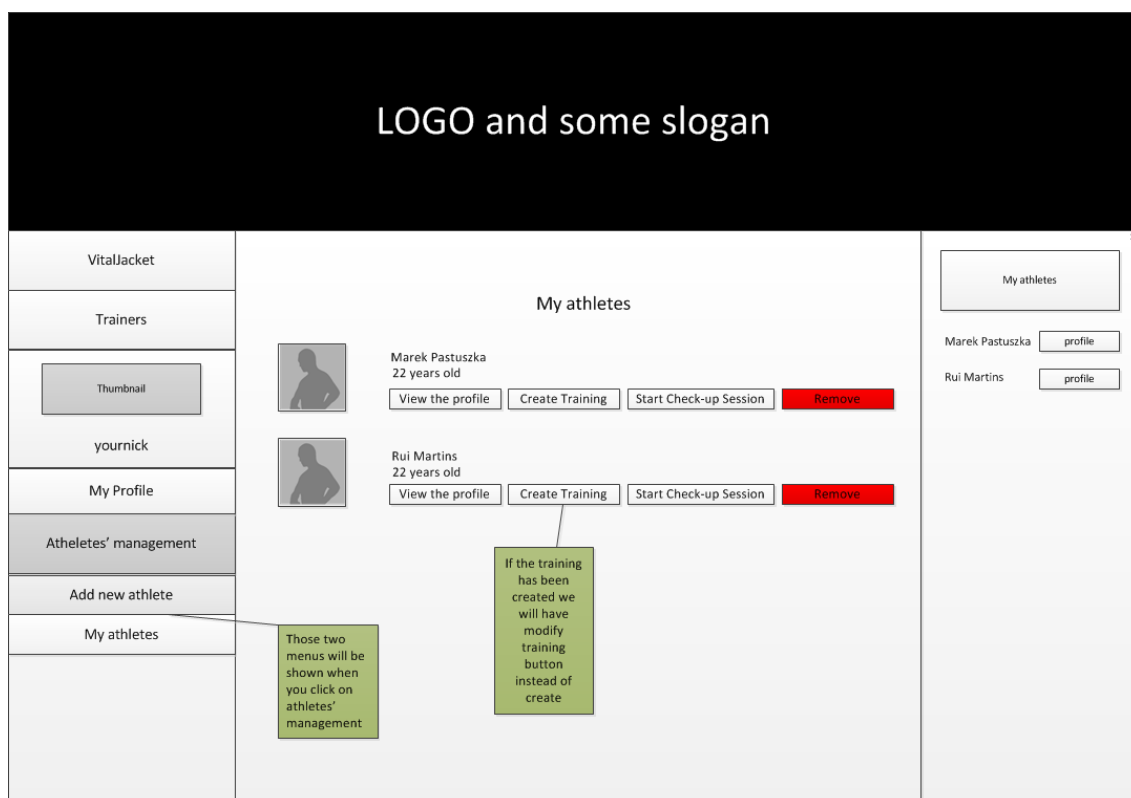


Edit Profile






Login screen



My Athletes

LOGO and some slogan		
VitalJacket	 <p>yournick 22 years old Active since: 04/04/2011</p> <p>Edit Profile</p>	Notes
Trainers		
<div>Thumbnail</div> <div>yournick</div>		
My Profile		
My training		
My Check-up Sessions		

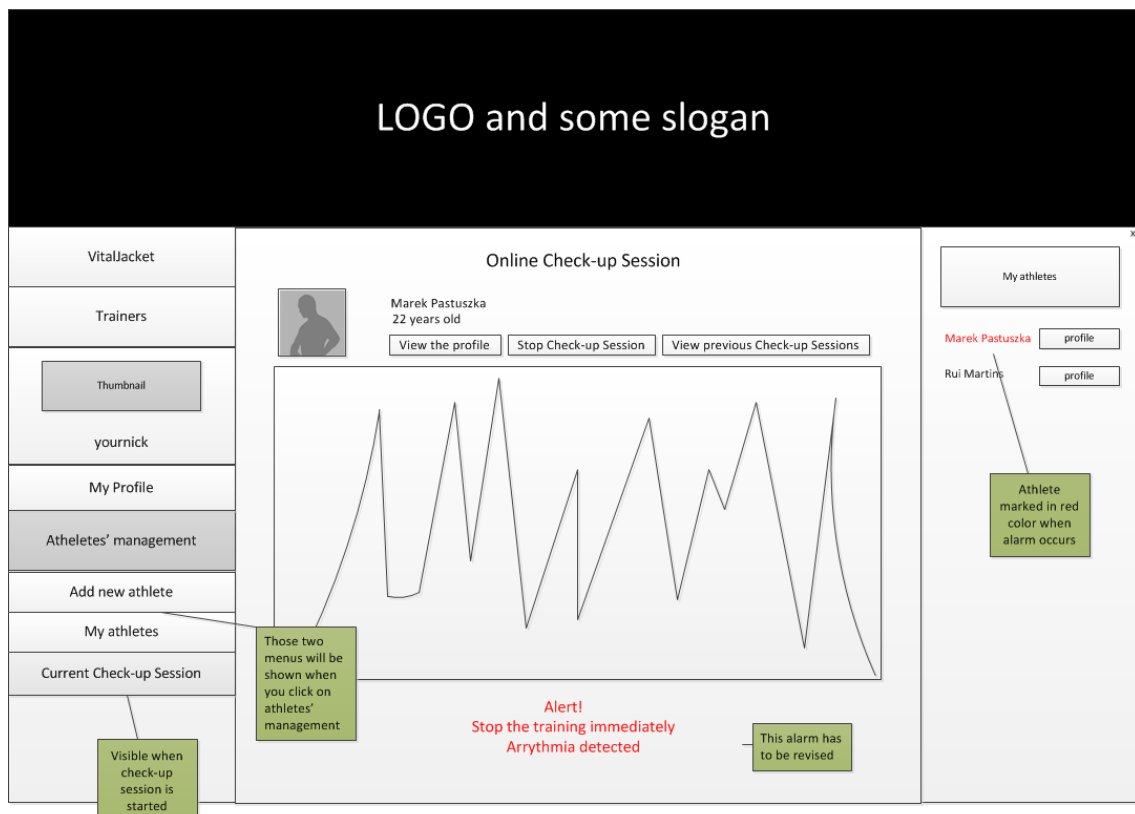
My Profile

LOGO and some slogan		
VitalJacket	Currently assigned training:	Notes
Trainers		
<div>Thumbnail</div> <div>yournick</div>		
My training		
My Check-up Sessions		

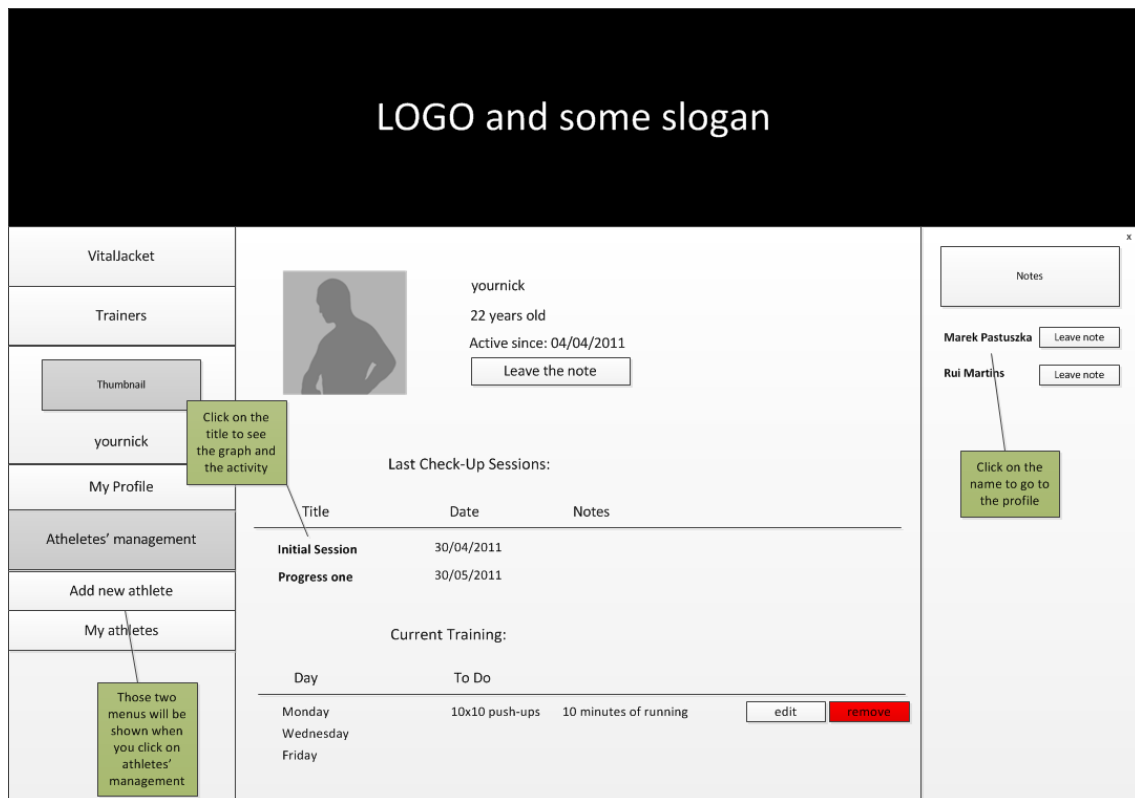
My Training



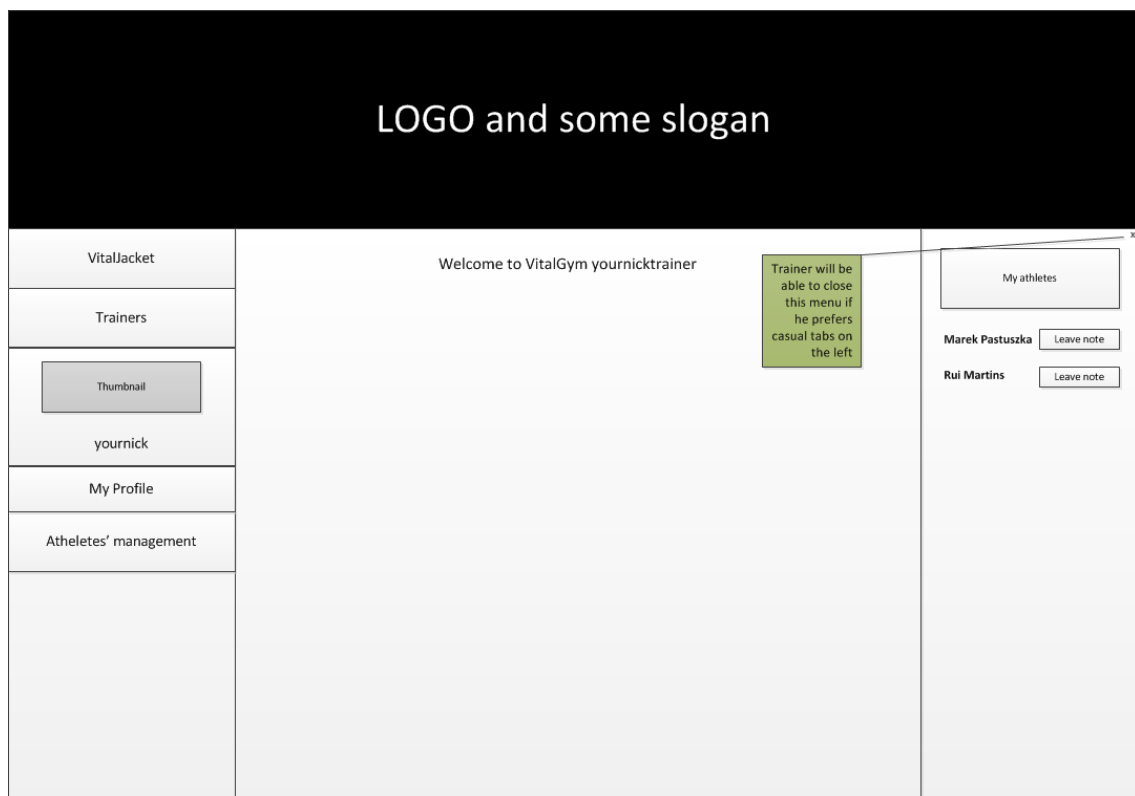
Online check-up



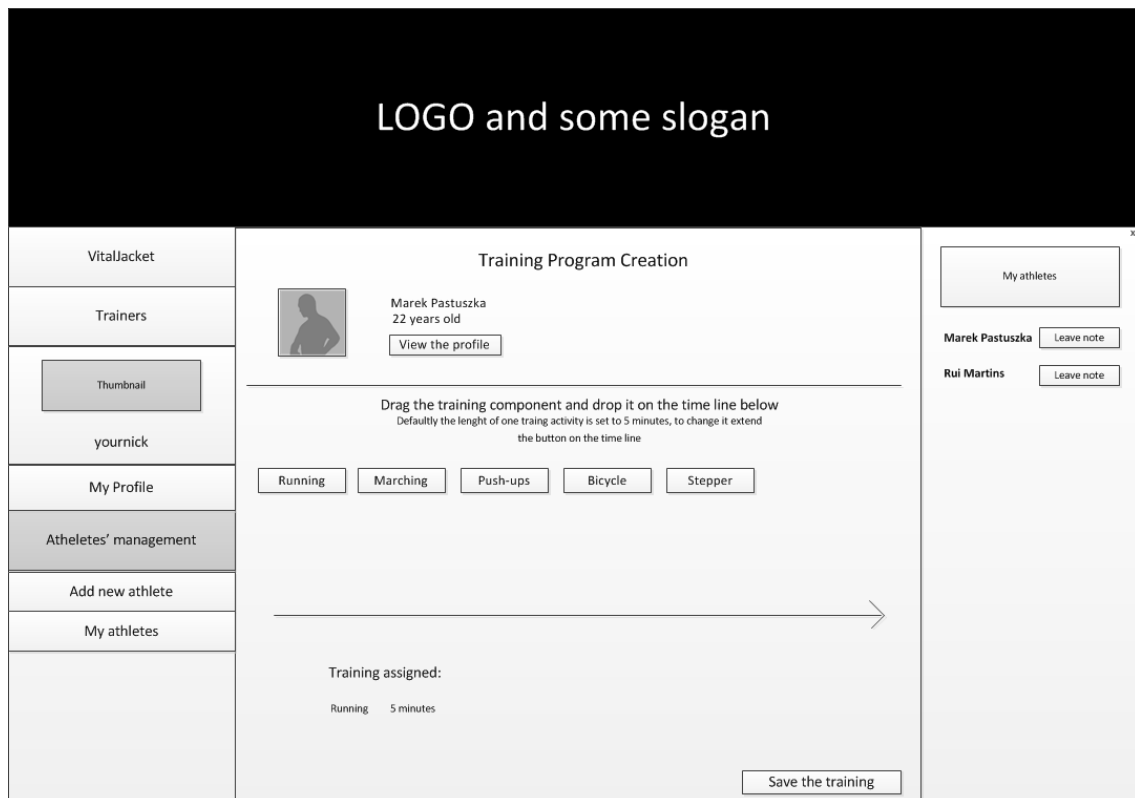
Online check-up Trainer



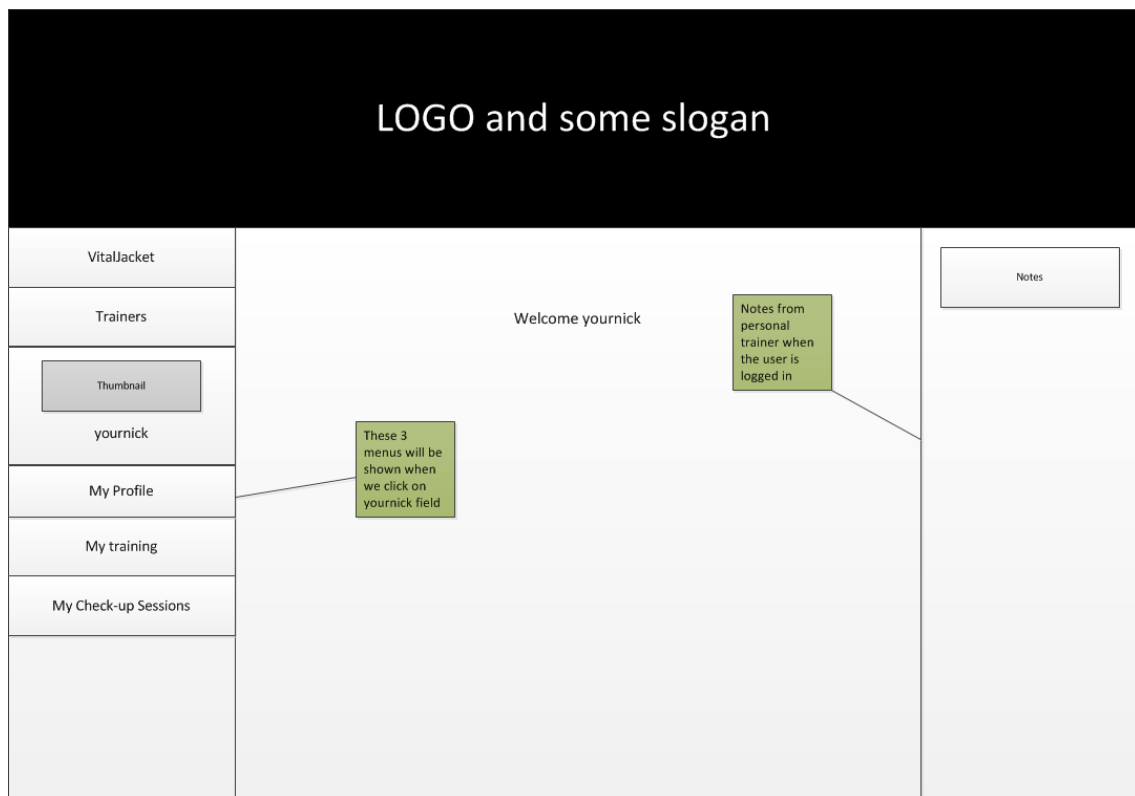
Personal Trainer view of Athlete Profile



Trainer initial screen



Training Program Creation



When logged in

Code and development documentation

This section intends to serve as a guide to the classes that team members create. The team members should comment their code with *javadoc* and the generated web pages will be included in this section. The use of the class functions to obtain a certain result should also be included if necessary.

Javadoc

Most of the documentation will be in javadoc format.

[Here](#) is the generated (mostly complete) javadoc for [r224](#) (revisions 224).

Other code

As not every piece of code in the project is java (and therefore no javadoc) we'll document that code here.

HTML/JavaScript

live.html

This code is mainly a JavaScript function that gets data from an xml feed (described in the JSP section). This [flot](#) library is used to render a graph of ECG data.

offline.html

This is similar to the live.html but has to handle the navigation through the ECG data. The main differences are:

- There is an initial setup, in order to choose which file is read (the offline session is stored in a file).
- The notion of time is important (in the live counterpart the time is not displayed).
- The navigation through the ECG is done by requesting new data in the feed, while changing the labels if necessary (e. g. if the user wants to see a minute of data, the graph should not display labels in milliseconds).

JSP

live.jsp

Requests data from the live data Bean, and puts it in an xml feed with this format:

```
<feed>

  <heartrate>
    YYY
  </heartrate>

  <values>
    <v>ZZZ</v>
    ...
    <v>YYZ</v>
  </values>
</feed>
```

offline_req.jsp

Calls the method startReading in the offline data bean.

offline.jsp

Requests data from the offline data bean, sending the beginning and the end of the sample desired. The provided feed is like this:

```
<values>

  <v>ZZZ</v>
  ...
  <v>YYZ</v>
</values>
```

JSF (XHTML)

layout.xhtml

This layout.xhtml loads the layout of the page, and, depending on the type of user that is logged in, loads the appropriate menu.

centerContent.xhtml

This file loads the content of the page depending on the state of the navigationManagedBean.

Navigation is done by calling the function navigationManagedBean.goTo(), this changes the state of the object, and as the page is refreshed, different content is loaded. Using a bean to handle navigation allows us to implement authorization easily, because that bean can also see what is the type of user is, and choose to allow or disallow the navigation, instead redirecting the user to an error page, this however is not implemented.